OLD DOMINION UNIVERSITY RESEARCH FOUNDATION

DEPARTMENT OF ELECTRICAL ENGINEERING
SCHOOL OF ENGINEERING
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

DESIGN OF INFRASOUND-DETECTION SYSTEM VIA ADAPTIVE LMSTDE ALGORITHM

By

Camille S. Khalaf

and

John W. Stoughton, Principal Investigator

Final Report
For the period ending December 31, 1984

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Research Contract NAS1-17099
Task Authorization No. 33
Dr. Allan J. Zuckerwar, Technical Monitor
IRD-Acoustics & Vibration Instrumentation Section

November 1984

DEPARTMENT OF ELECTRICAL ENGINEERING
SCHOOL OF ENGINEERING
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

DESIGN OF INFRASOUND-DETECTION SYSTEM VIA ADAPTIVE LMSTDE ALGORITHM

By

Camille S. Khalaf

and

John W. Stoughton, Principal Investigator

Final Report
For the period ending December 31, 1984

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Research Contract NAS1-17099
Task Authorization No. 33
Dr. Allan J. Zuckerwar, Technical Monitor
IRD-Acoustics & Vibration Instrumentation Section

Submitted by the
Old Dominion University Research Foundation
P.O. Box 6369
Norfolk, Virginia 23508

November 1984

ABSTRACT

DESIGN OF INFRASOUND-DETECTION SYSTEM
VIA ADAPTIVE LMSTDE ALGORITHM

Camille S. Khalaf
Old Dominion University
Director: John W. Stoughton

A proposed solution to an aviation safety problem is based on passive detection of turbulent weather phenomena through their infrasonic emission. This thesis describes a system design that is adequate for detection and bearing evaluation of infrasounds. An array of four sensors, with the appropriate hardware, is used for the detection part. Bearing evaluation is based on estimates of time delays between sensor outputs. The generalized cross correlation (GCC), as the conventional time-delay estimation (TDE) method, is first reviewed. An adaptive TDE approach, using the least mean square (LMS) algorithm, is then discussed. A comparison between the two techniques is made and the advantages of the adaptive approach are listed. The behavior of the GCC, as a Roth processor, is examined for the anticipated signals. It is shown that the Roth processor has the desired effect of sharpening the peak of the correlation function. It is also shown that the LMSTDE technique is an equivalent implementation of the Roth processor in the time domain. A LMSTDE lead-lag model, with a variable stability coefficient and a convergence criterion, is designed. This model is employed in an automatic scheme developed for the sensor array. The software and hardware system parameters are derived and determined. The effectiveness of the system is illustrated through simulation and field testing.

ACKNOWLEDGEMENTS

I wish to thank my thesis advisor, Dr. John W. Stoughton, for his invaluable assistance and insights, and for his support and understanding when I needed them most. I wish to acknowledge my gratitude to Dr. Allan J. Zuckerwar, Acoustical and Vibrational Instrumentation Section (AVIS), NASA/Langley Research Center, whose constant effort and assistance made this research possible. My thanks also go to Mr. Harlan K. Holmes, Head-AVIS, Mr. Ralph Lewis, Aerospace Technician-AVIS, and Mr. David Katzoff, Governor's School, for their time and effort. I acknowledge the financial support by the National Aeronautics and Space Administration under contract NAS1-17099, Tasks 10 and 33.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES - Concluded

DEFINITION OF SYMBOLS

t	continuous time
n	discrete time
τ	time lag
τ_d	time delay
f	frequency in Hz
f_s	sampling rate
Δt	sampling period
ω	radian frequency
D	discrete delay
θ	bearing relative to microphones axis
β	bearing relative to North
ϕ	elevation angle
$\Delta\theta$	bearing resolution
ℓ	distance between two sensors
v	sound velocity
K_s	stability constant
C	scale of stability constant
N	filter length
L	record length
M	file length
K	variance window
VT	variance threshold

$S(\cdot)$	signal
$m(\cdot)$	noise
$w(\cdot)$	weighting
$\delta(\cdot)$	delta function
$x(\cdot)$	input signal
$R_i(n)$	random sequence
$\varepsilon(n)$	filter error function
$d(n)$	filter desired response
$y(n)$	filter output
$X(n)$	filter input vector
$R_{x_i x_j}(\cdot)$	correlation function
$G_{x_i x_j}(\cdot)$	power spectrum
$V_{x_i x_j}(\cdot)$	covariance matrix
$\gamma_{x_i x_j}(\cdot)$	coherence function
$F[\cdot]$	Fourier transform
$E[\cdot]$	expected value
$\nabla[\cdot]$	gradient
$\circledast$	convolution

CHAPTER 1

INTRODUCTION

1.1 Motivation

Current research programs have lead to significant advances in ground-based and airborne equipment for providing information relative to severe turbulent weather. However, most of these programs are still in their experimental stages and very few are operational. There continues to be a serious aviation safety problem associated with aircraft operations in the vicinity of severe storms.

In 1981, general aviation aircraft numbered more than 200,000 and flew more than 40 million hours [1]. General aviation operations accounted for 662 fatal accidents from all causes, with 1,265 fatalities (FAA, 1981). Informal accident cause/factor statistics from the National Transportation Safety Board for 1981 indicate that weather caused, or was a factor, in 40 percent (289 cases) of the U. S. general aviation accidents. Earlier statistics indicated that turbulence is the largest single cause of weather-related air carrier accidents in the U. S. From 1962 to 1974, turbulence was either a cause of or a contributing factor in 189 of 450 weather-related cases [2].

The two types of turbulence usually encountered are clear-air turbulence (CAT) and thunderstorm-related turbulence. CAT, a problem for all aircraft, cannot be seen because it usually has no cloud signature. It may develop in a standing wave caused by air moving over mountainous

terrain, and is frequently associated with shear-induced Kelvin-Helmholtz atmospheric waves occurring in a statistically stable atmosphere [3]. Accidents caused by CAT are not as serious as the ones related to turbulence associated with thunderstorms. A CAT accident may result in discomfort, injuries, aircraft damage, and/or unscheduled landing. Thunderstorms and other convective clouds are critically important sources of low-altitude turbulence and wind variability. Many produce strong downdrafts that transport air downward, which then spreads out rapidly over the ground. This mechanism, if encountered during take off or landing of aircraft, may result in serious if not fatal accidents.

One of the pressing aviation safety problems is that of providing the pilot with information needed to avoid turbulence hazards which exceed the design capabilities of the airplane. A proposed solution to this problem through passive detection of turbulence is presented in the next section.

1.2 Turbulence Detection

The primary technique for detecting turbulence and storm cells is Doppler radar. This technique requires the presence of reflective particles such as precipitation or dust for meterological applications. Doppler radar proved ineffective in cases of CAT or developing storm cells due to the absence of reflective particles in these phenomena. The proposed technique consists of passive detection of large-scale patches of turbulence in the Earth's atmosphere through infrasonic emissions. The infrasonic technique offers the advantages of being (1)

passive, (2) inexpensive, and (3) inherently more sensitive to atmospheric disturbances than electromagnetic or optical techniques. The assumed utilization of the infrasonic technology is illustrated in Figure 1.1. Infrasonic emissions from the patch of turbulence are detected at two stations, each containing an array of four sensors. By means of real-time signal processing, the direction of the turbulent source, based on time delays between sensors, is determined at each station. The turbulence is then located by triangulation between the two stations. This information is relayed from a control center to a flight services advisory, and from there to the pilots of approaching aircraft. A series of such stations would provide early warning along the length of domestic air traffic routes.

It is informative at this point to mention that the first step toward realization of the infrasonic technique was the development of a unified acquisition system for acoustic data [4] by Dr. Allan J. Zuckerwar and Mr. Harlan K. Holmes at NASA Langley Research Center. This system served an important role in understanding the detection problem during the early stage of this research.

The delay-to-angle conversion needed in this passive approach is illustrated in Figure 1.2. If τ_d is the time delay between the arrival of the infrasonic signal at two spatially separated sensors, then the direction of the turbulent source is given by

$$\theta = \cos^{-1}\left(\frac{\tau_d \times V}{\ell}\right) \qquad (1.1)$$

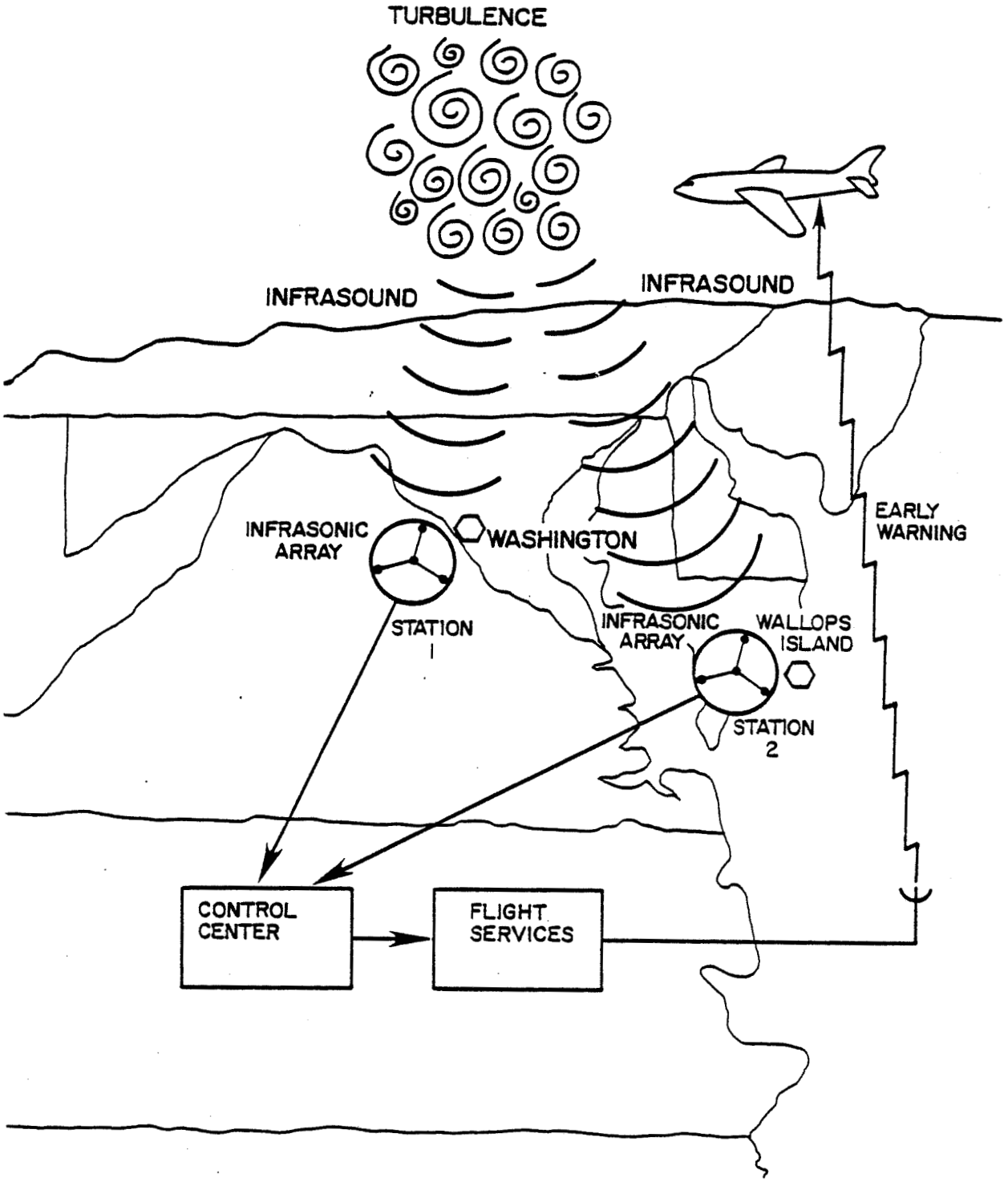


Figure 1.1. Early warning system.

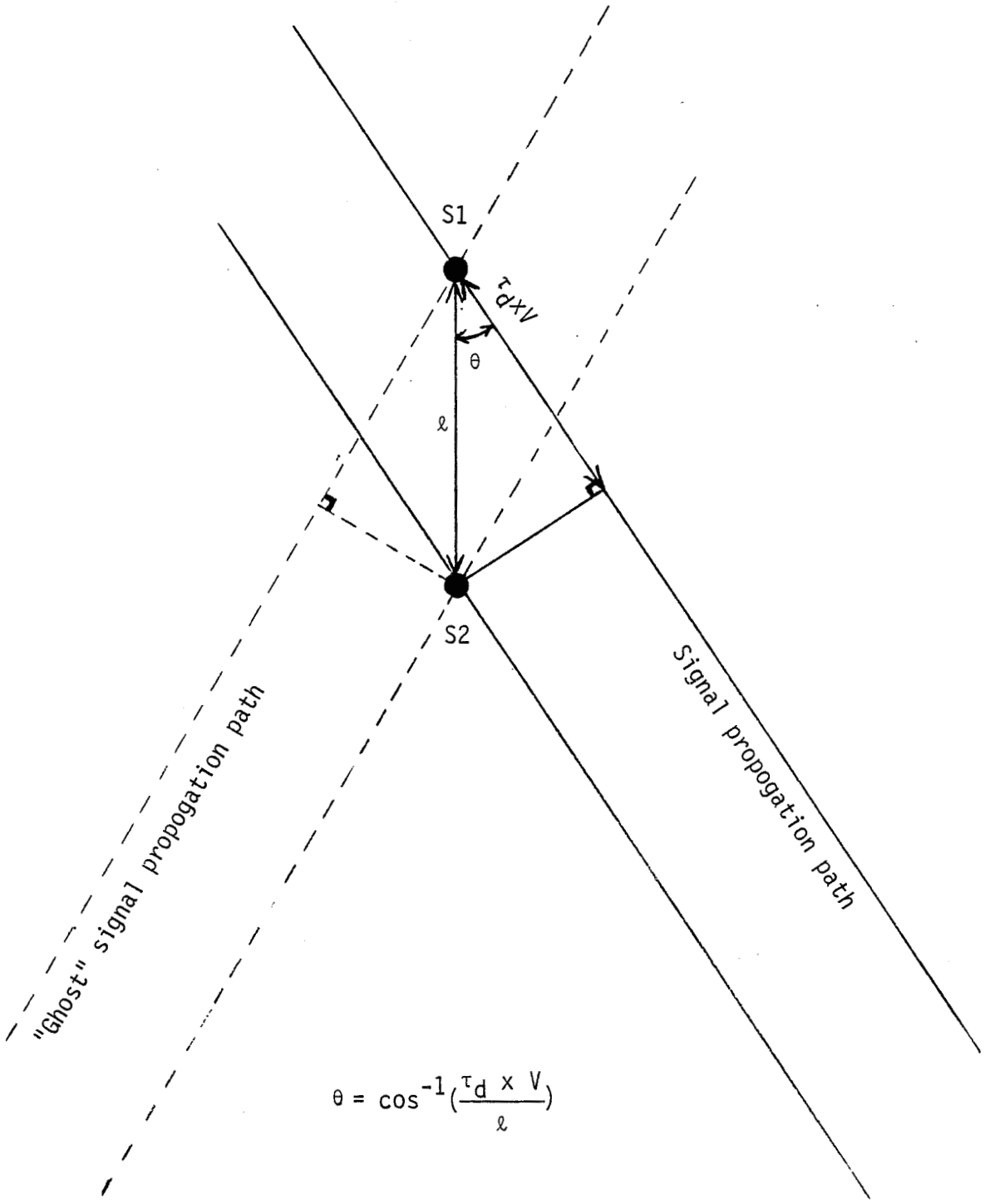


Figure 1.2. Delay-to-angle conversion.

where V is the sound velocity (1080 ft/sec at 0°C), and ℓ is the distance between the two sensors. For a given delay, the solution of equation (1.1) can be interpreted in two ways as shown in the figure. The signal associated with the false direction is usually referred to as the "ghost" signal. The presence of the ghost-signal requires a minimum of three sensors in the array to uniquely determine the source direction.

1.3 Research Objectives

The infrasonic technique, proposed to provide pilots with an early warning of turbulence, was described in the previous section. The three aspects that are essential to its success are the following:

1. emission and propagation of infrasounds from turbulent weather phenomena;
2. detection of infrasounds and bearing evaluation of their sources at every station;
3. locating the sources by triangulation between stations via a communication network that is also responsible for transferring the information to pilots.

The first aspect is out of the designer's control and is based on the theoretical predictions of infrasonic emission and propagation. The third aspect constitutes the commercial phase of the program once the performance of the individual stations proves satisfactory.

The goal of this research is the overall system design that is adequate for implementing the second aspect of the infrasonic technique. In this context, the system should function as an infrasonic detector,

time-delay estimator, and delay-to-angle converter. The detection part involves the sensor array and corresponding hardware blocks while the delay-to-angle conversion is accomplished by a simple software routine. The main design issue is that of an appropriate signal processing technique for time-delay estimation. The GCC method and the LMS parametric technique will both be reviewed. More specifically, the GCC method using a Roth weighting function, and the adaptive implementation of the LMS technique, will be examined. Design decisions, regarding the TDE technique as well as the hardware system, will then be made according to the anticipated signal characteristics.

In Chapter 2, the theory behind TDE techniques is reviewed. Based on this theory and the anticipated infrasounds, the system design is presented in Chapter 3. Chapter 4 evaluates the system through simulation of the basic TDE algorithm response and discussion of the system performance in the actual field. Conclusions and future research are presented in Chapter 5.

CHAPTER 2

THEORY

2.1 Introduction

The problem identified in chapter one is a fundamental passive sonar signal processing problem in which delays between the times of arrival of the pertinent acoustic waves at four sensors are to be estimated. This chapter will discuss the theory behind TDE techniques. Section two reviews the conventional generalized cross correlation approach (GCC) while section three reviews the Roth processor specifically. Of particular interest is a parametric approach through Widrow's adaptive filter. The filter structure is reviewed in section four while its application to TDE is presented and thoroughly investigated in the fifth section. The chapter is concluded in section six by presenting the advantages of the adaptive least mean squared time delay estimation (LMSTDE) method.

2.2 Generalized Cross Correlation Approach

The GCC approach to time delay estimation has been discussed by many investigators. Well known references include papers written by Knapp and Carter [5], and Hassab and Boucher [6]. This section only reviews this approach through Figure 2.1. $x_1(t)$ and $x_2(t)$ are sampled at two spatially separated sensors and then fed to a basic cross correlator. The basic cross correlator, as discussed by Papoulis [7],

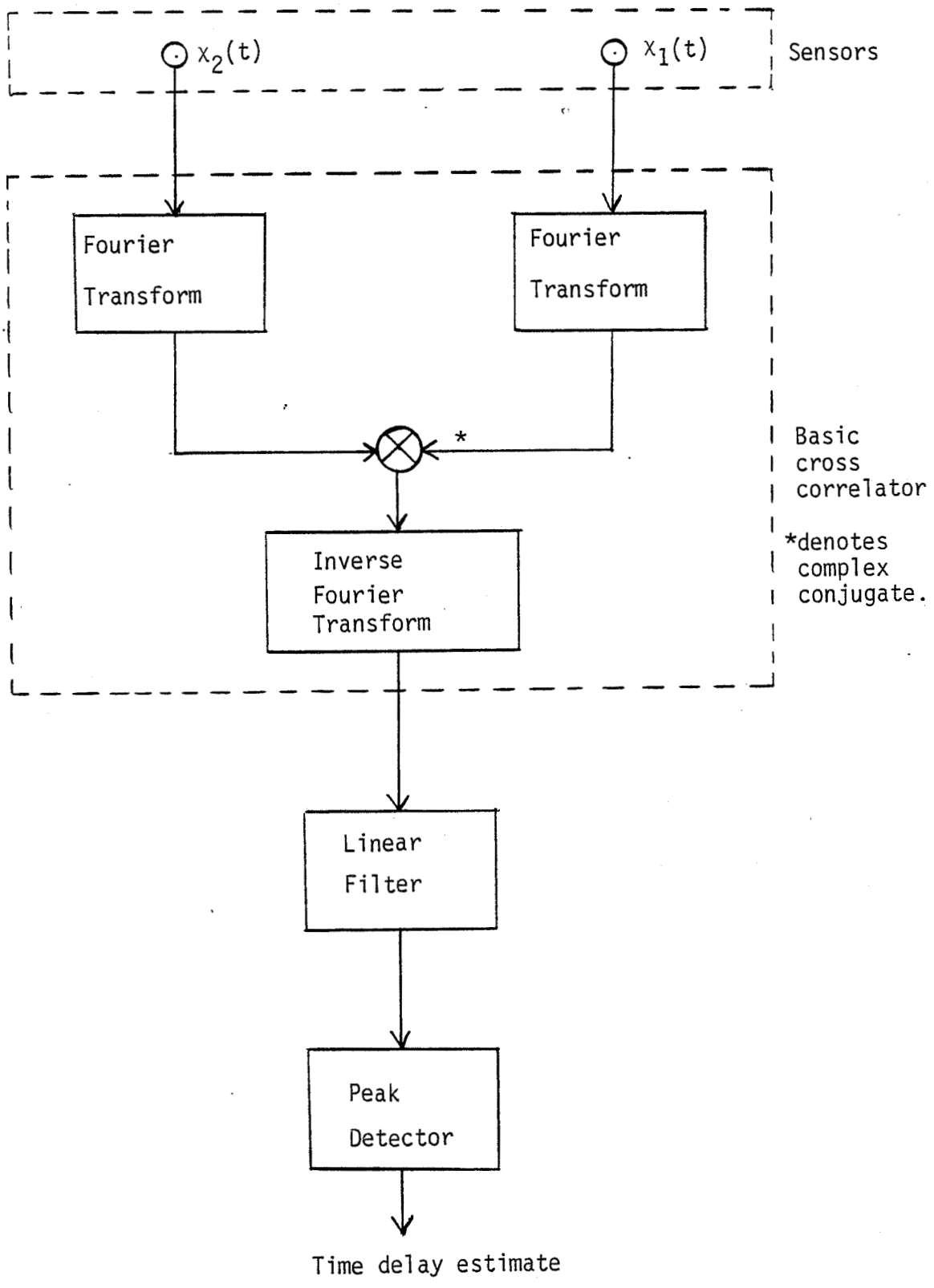


Figure 2.1. Generalized cross correlation.

computes the cross correlation function, $R_{x_1x_2}(\tau)$, between $x_1(t)$ and $x_2(t)$ by means of the inverse Fourier transform of their cross power spectrum, $G_{x_1x_2}(f)$. Then the delay estimate is simply the abscissa value at which the cross correlation function peaks. In order to improve the accuracy of the delay estimate, a linear filter, $w(t)$, is convolved with the output of the cross correlator. A peak detector routine is then used to determine the abscissa value of the peak in the filtered correlation function. In practice, a frequency weighting $W(f) = F[w(t)]$, where $F[\cdot]$ denotes the Fourier transform of $[\cdot]$, which is equivalent to $w(t)$, is applied to the cross power spectrum prior to taking the inverse Fourier transform. This frequency weighting replaces the linear filter so that all computations, except for the peak detector routine, are done in the frequency domain. A discussion of this weighting, after mathematically modeling the system, is presented next.

A signal emanating from a remote source and monitored in the presence of noise at two spatially separated sensors can be mathematically modeled as

$$x_1(t) = S_1(t) + m_1(t) \tag{2.1a}$$

$$x_2(t) = \alpha S_1(t + \tau_d) + m_2(t) \tag{2.1b}$$

where $S_1(t)$, $m_1(t)$ and $m_2(t)$ are real, zero-mean, jointly stationary random processes, τ_d denotes time delay and α an attenuation factor. Signal $S_1(t)$ is assumed to be uncorrelated with $m_1(t)$ and $m_2(t)$.

The cross correlation between $x_1(t)$ and $x_2(t)$ is related to the cross power spectral density function, $G_{x_1x_2}(f)$, by the Fourier transform relationship

$$R_{x_1x_2}(\tau) = \int_{-\infty}^{\infty} G_{x_1x_2}(f)\ e^{j2\pi f\tau}\ df \qquad (2.2)$$

In practice, only an estimate $G'_{x_1x_2}(f)$ of (2.2) can be obtained from finite observations of $x_1(t)$ and $x_2(t)$. Consequently, the output of the basic cross correlator, when no weighting is used, is:

$$R'_{x_1x_2}(\tau) = \int_{-\infty}^{\infty} G'_{x_1x_2}(f)\ e^{j2\pi f\tau}\ df \qquad (2.3)$$

It is informative at this point to examine the shape of the cross correlation function $R_{x_1x_2}(\tau)$, so that the necessity of using a weighting function can be justified. An expression for $R_{x_1x_2}(\tau)$ can be obtained from (2.1), using the expectation operator $E[\cdot]$, as

$$R_{x_1x_2}(\tau) = E[x_1(t)\ x_2(t-\tau] = \alpha\ R_{s_1s_1}(\tau-\tau_d) + R_{m_1m_2}(\tau) \qquad (2.4)$$

The Fourier transform of (2.4) gives the cross power spectrum

$$G_{x_1x_2}(f) = \alpha G_{s_1s_1}(f)\ e^{-j2\pi f\tau_d} + G_{m_1m_2}(f)\ . \qquad (2.5)$$

If $m_1(t)$ and $m_2(t)$ are uncorrelated, $G_{m_1 m_2}(f) = 0$, and since multiplication in one domain is a convolution in the transformed domain, it follows from (2.5) that

$$R_{x_1 x_2}(\tau) = \alpha R_{s_1 s_1}(\tau) \circledast \delta(\tau-\tau_d) \qquad (2.6)$$

where $\circledast$ denotes convolution and $\delta(\tau-\tau_d) = F^{-1}\left[e^{-j2\pi f\tau_d}\right]$. One interpretation of (2.6) is that the delta function has been spread or "smeared" by the Fourier transform of the signal spectrum. If $S_1(t)$ is a white noise source, then its autocorrelation function is a delta function and no spreading takes place. However, for most practical applications this is not the case and spreading acts to broaden the peak of the cross correlation.

To minimize the spreading effect, many weighting functions have been proposed in the literature (see Table 2.1) to operate on the cross power spectrum given in (2.5). With a general weighting, $Wg(f)$, the estimate of the generalized cross correlation becomes

$$R^g_{x_1 x_2}(\tau) = \int_{-\infty}^{\infty} Wg(f)\, G_{x_1 x_2}(f)\, e^{j2\pi f\tau}\, df \qquad (2.7)$$

$Wg(f)$ should be chosen to ensure a large sharp peak in $R^g_{x_1 x_2}(\tau)$ rather than a broad one in order to obtain a good time-delay resolution.

2.3 Roth Processor

The weighting functions found in the literature are listed in Table

2.1, where the notation $\gamma(f)$ has been used for the coherence function which is defined as

$$\gamma_{12}(f) = \frac{G^2_{x_1x_2}(f)}{G_{x_1x_1}(f)\ G_{x_2x_2}(f)}$$

The selection of $Wg(f)$ to optimize certain performance criteria has been studied by several investigators (see, for example [8]). The purpose of considering the Roth processor is that it is equivalent to the time domain weighting involved in the adaptive LMSTDE approach. This equivalence will be shown in section five and will be helpful in gaining insight to the adaptive method.

The weighting proposed by Roth [9] is

$$W_R(f) = \frac{1}{G_{x_1x_1}(f)}$$

Substituting for $Wg(f)$ in (2.7) yields

$$R_{x_1x_2}(\tau) = \int_{-\infty}^{\infty} \frac{G_{x_1x_2}(f)}{G_{x_1x_1}(f)}\ e^{j2\pi f\tau}\ df \qquad (2.8)$$

Equation (2.8) estimates the impulse response of the optimum linear (Wiener-Hopf) filter [10]

$$H(f) = \frac{G_{x_1x_2}(f)}{G_{x_1x_1}(f)}$$

Table 2.1. Weighting Functions

Processor Name	Weight	Reference
Cross Correlation	1	[8], [7]
Roth Processor	$\frac{1}{G_{x_1 x_1}(f)}$	[9]
SCOT[1]	$\frac{1}{G_{x_1 x_1}(f)\, G_{x_2 x_2}(f)}$	[10]
PHAT[2]	$\frac{1}{\lvert G_{x_1 x_2}(f)\rvert}$	[10]
Eckart Filter	$\frac{1}{G_{m_1 m_1}(f)\, G_{m_2 m_2}(f)}$	[5]
ML[3]	$\frac{\lvert\gamma_{12}(f)\rvert^2}{\lvert G_{x_1 x_2}(f)\rvert[1-\lvert\gamma_{12}(f)\rvert^2]}$	[5]

[1] Smoothed coherence transform

[2] Phase transform

[3] Maximum likelihood

which "best" approximates the mapping of $x_1(t)$ to $x_2(t)$. If $m_1(t) \neq 0$, then $G_{x_1 x_1}(f) = G_{S_1 S_1}(f) + G_{m_1 m_1}(f)$, and equation (2.6) becomes for the Roth processor

$$R_{x_1 x_2}(\tau) = \delta(\tau - \tau_d) \circledast \int_{-\infty}^{\infty} \frac{\alpha G_{S_1 S_1}(f)}{G_{S_1 S_1}(f) + G_{m_1 m_1}(f)} e^{j2\pi f \tau} \, df \quad (2.9)$$

From (2.9) we can conclude that only when $G_{m_1 m_1}(f)$ is negligible or when it equals any constant times $G_{S_1 S_1}(f)$, the spreading does not occur and $R_{x_1 x_2}(\tau)$ becomes a delta function. However, as can be seen from the integral part in (2.9), the Roth processor has the desirable effect of suppressing those frequency regions where $G_{m_1 m_1}(f)$ is large and $G_{x_1 x_2}(f)$ is more likely to be in error.

The generalized cross correlation approach suffers regardless of which weighting function is used, from two basic facts. First, it relies on a sequence of fast Fourier transform (FFT) computations that tends to be time consuming. Second, as seen in Table 2.1, it requires a priori knowledge of the signal and noise statistics to implement the specific weighting. In passive detection problems, this information is unknown and if it were to be estimated, it would increase the complexity of the process and the time involved. The next section presents the adaptive LMSTDE algorithm, which is equivalent to the Roth processor, and is able to overcome both difficulties found in the generalized cross correlation method.

2.4 Adaptive Filters

This section will review the basic structure of adaptive filters discussed by Widrow [11], while section 2.5 presents the application of the filter in TDE problems.

A signal filtering approach using an adaptive filter is in some sense a self-designing (really self-optimizing) process. The adaptive filter described here bases its own "design" (its internal adjustment settings) upon estimated or measured statistical characteristics of the input and output signals. The statistics are not measured explicitly; rather, the filter design is accomplished in a single process by a recursive LMS algorithm that automatically updates the system coefficients with the arrival of each set of data samples. Figure 2.2 illustrates schematically the adaptive filter used in this case as a linear combinatorial system. The filter consists of a set of variable weights (filter coefficients) whose input are the sampled input signals, a summer to add the weighted signals, and an algorithm to adjust the weights automatically. The impulse response of such a discrete system is completely controlled by the weight settings. The adaptation process automatically seeks an optimal filter impulse response by adjusting the weights using gradient techniques to minimize the mean-square-error function.

The analysis of the adaptive filter can be developed by assuming that the input signals are statistically stationary random processes. Let the nth set of input signals be a vector $X(n)$ of length N,

$$X^T(n) = [x_1(n) \quad x_2(n) \quad \ldots \quad x_N(n)]$$

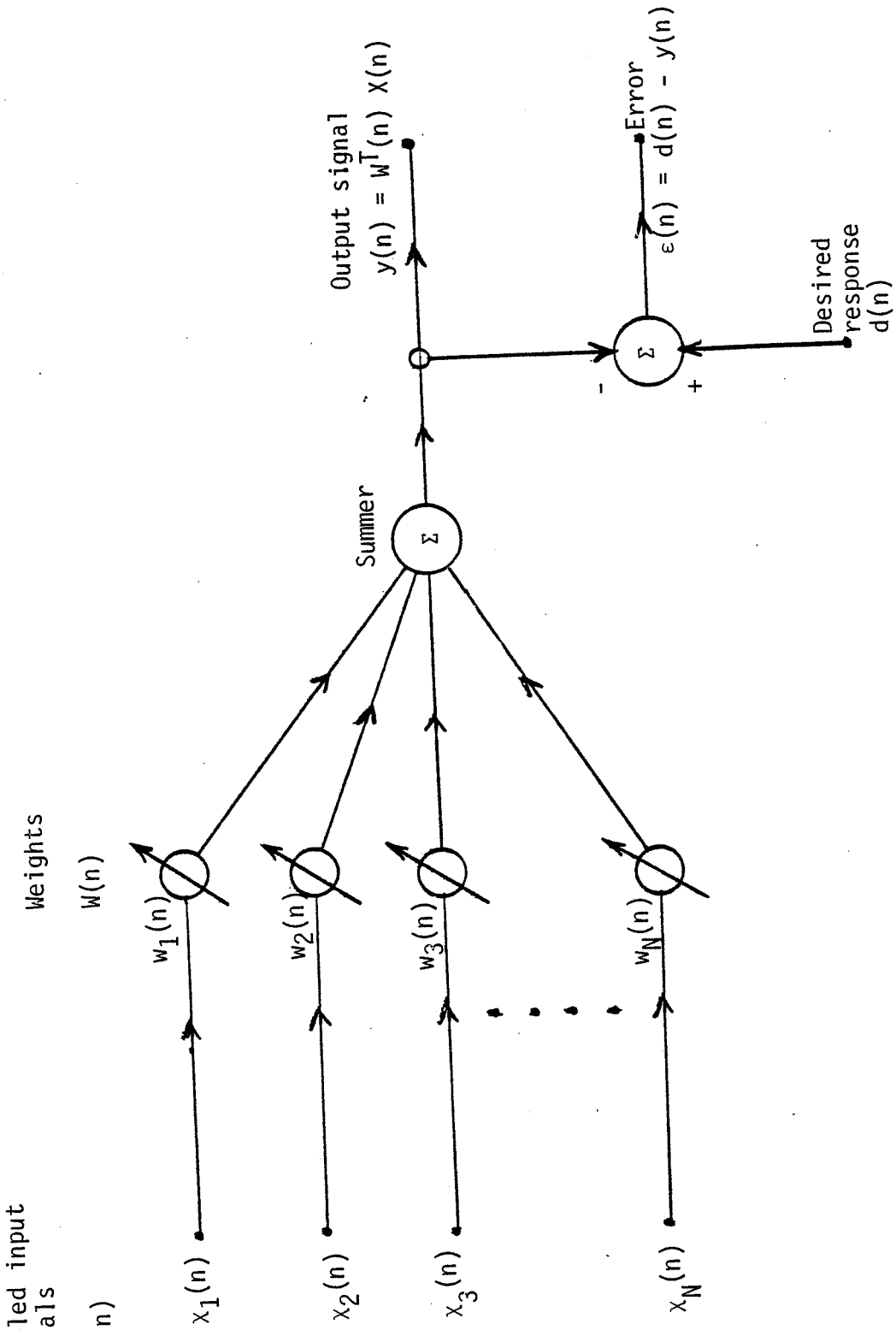


Figure 2.2. Adaptive linear combinatorial system.

where X^T denotes the transpose of X. Let the set of weights, at the nth time, be designated by the vector $W^T(n) = [w_1(n) \quad w_2(n) \ \ldots \ w_N(n)]$. The nth output signal is $y(n) = \sum_{i=1}^{N} w_i(n)\, x_i(n)$. This can be written in matrix form as

$$y(n) = W^T(n)\, X(n) = X^T(n)\, W(n). \tag{2.10}$$

Denoting the desired response by $d(n)$, the error at the nth time is

$$\varepsilon(n) = d(n) - y(n) = d(n) - W^T(n)\, X(n). \tag{2.11}$$

The square of the error is

$$\varepsilon^2(n) = d^2(n) - 2d(n)\, X^T(n)\, W(n) + W^T(n)\, X(n)\, X^T(n)\, W(n). \tag{2.12}$$

The expected value of $\varepsilon^2(n)$ is

$$E\,[\varepsilon^2(n)] = \overline{d^2}(n) - 2\, V_{dx}^T\, W(n) + W^T(n)\, V_{xx}\, W(n). \tag{2.13}$$

where $\overline{(\cdot)}$ denotes the mean value of $(\cdot)$, or $E[\cdot]$. The vector V_{dx} is the cross covariance between $d(n)$ and $x(n)$ and is defined as

$$V_{dx} = E[d(n)\, X(n)] = E \begin{bmatrix} d(n)\, x_1(n) \\ d(n)\, x_2(n) \\ \vdots \\ d(n)\, x_N(n) \end{bmatrix} \tag{2.14}$$

and V_{xx} denotes the auto-covariance matrix of $X(n)$

$$V_{xx} = E\,[X(n)\ X^T(n)] = E\begin{bmatrix} x_1(n)\ x_1(n) & x_1(n)\ x_2(n)\ldots \\ x_2(n)\ x_1(n) & x_2(n)\ x_2(n)\ .. \\ \bullet & \\ \bullet & \\ \bullet & \\ & X_N(n)\ X_N(n) \end{bmatrix} \tag{2.15}$$

It may be observed from (2.13) that for stationary input signals, the mean-square error is a second-order function of the weights. Thus, the mean-square-error function may be viewed, as suggested by Widrow [11], as a "performance surface" for the adaptive process that has a unique stationary point (minimum) which can be sought using gradient techniques.

The gradient at any point on the performance surface can be obtained by differentiating the mean-square-error function (2.13) with respect to the weights. The gradient is

$$\nabla[\overline{\varepsilon^2}(n)] = -2\ V_{dx} + 2\ V_{xx}\ W(n) \tag{2.16}$$

The "optimal" weight vector, W_{LMS}, that yields the least-mean-square error, is found where the gradient is zero. Accordingly,

$$V_{dx} = V_{xx}\ W_{LMS}$$

or

$$W_{LMS} = (V_{xx})^{-1}\ V_{dx} \tag{2.17}$$

where $[\cdot]^{-1}$ denotes the inverse of $[\cdot]$. Equation (2.17) is the Wiener-Hopf equation in matrix form [12]. Solving (2.17) for the optimum weight vector (the minimum point on the performance surface) can present serious computational problems. However, it will be shown that the adaptation process tries to find an exact or an approximate solution to the Wiener-Hopf equation by using the LMS algorithm with less computational complexity.

When using the LMS algorithm, changes in the weight vector are made along the direction of the estimated gradient vector. Accordingly,

$$W(n+1) = W(n) + K_s \nabla [\overline{\varepsilon^2}(n)] \tag{2.18}$$

where

$W(n) \triangleq$ weight vector before adaptation

$W(n+1) \triangleq$ weight vector after adaptation

$K_s \triangleq$ scalar constant controlling the rate of convergence and stability ($K_s < 0$)

$\nabla'[\overline{\varepsilon^2}(n)] \triangleq$ estimate of gradient of $E[\varepsilon^2] = \overline{\varepsilon^2}$ with respect to W, with $W = W(n)$

One method for obtaining the estimated gradient of the mean-square-error function is to take the gradient of a single time sample of the squared error. That is,

$$\nabla'[\overline{\varepsilon^2}(n)] = \nabla\ [\varepsilon^2(n)] = 2\ \varepsilon(n)\ \nabla[\varepsilon(n)] \tag{2.19}$$

From equation (2.11),

$$\nabla[\varepsilon(n)] = \nabla\ [d(n) - W^T(n)\ X(n)] = -\ X(n)$$

Thus,

$$\nabla[\overline{\varepsilon^2}(n)] = -\ 2\ \varepsilon(n)\ X(n) \tag{2.20}$$

It can be shown that the gradient estimate of (2.20) is unbiased, so that

$$E\ \{\ \nabla\ [\overline{\varepsilon^2}(n)]\ \} = \nabla\ [\overline{\varepsilon^2}(n)]$$

Substituting (2.20) in (2.18) yields

$$W(n+1) = W(n) - 2\ K_s\ \varepsilon(n)\ X(n) \tag{2.21}$$

and the next weight vector is obtained by adding to the present weight vector the input vector scaled by the value of the error. This is the LMS algorithm.

Next, the expected value $E[W(n)]$ of the weight vector after a large number of iterations will be shown to converge to the Wiener solution given by (2.17). For this purpose, assume that the time between successive iterations of the algorithm is sufficiently long so that the input vectors $X(n)$ and $X(n+1)$ are uncorrelated. Taking the expected

value of both sides of (2.21) we obtain a difference equation in the expected value of the weight vector:

$$E[W(n+1)] = E[W(n)] - 2 K_s E\{ X(n) [d(n) - X^T(n) W(n)]\}$$

$$= [I+2 K_s V_{xx}] E[W(n)] - 2 K_s V_{dx} \qquad (2.22)$$

where I is an identity matrix. With an initial weight vector $W(0)$, $n+1$ iterations of (2.22) yield

$$E[W(n+1)] = [I+2 K_s V_{xx}]^{n+1} W(0) - 2 K_s \sum_{i=0}^{n} [I+2 K_s V_{xx}]^i V_{dx} \qquad (2.23)$$

Equation (2.23) can be put in diagonal form by using the normal-form expansion of the matrix V_{xx}. That is,

$$V_{xx} = Q^{-1} A Q$$

where the diagonal matrix of eigenvalues is A, and the square matrix of eigenvectors is the modal matrix Q. Equation (2.23) may now be expressed as

$$E[W(n+1) = [I+2 K_s Q^{-1}AQ]^{n+1} W(0) - 2 K_s \sum_{i=0}^{n} [I+2 K_s Q^{-1}AQ]^i V_{dx}$$

$$= Q^{-1}[I+2 K_s A]^{n+1} QW(0) - 2 K_s Q^{-1} \sum_{i=0}^{n} [I+2 K_s A]^i Q V_{dx}. \qquad (2.24)$$

Consider the diagonal matrix $[I+2\ K_s A]$. As long as its diagonal terms are all of a magnitude less than unity,

$$\lim_{n \to \infty} [I+ 2\ K_s A]^{n+1} \to 0$$

and the first term of (2.24) vanishes as the number of iterations increases. The summation factor in the second term of (2.24) becomes

$$\lim_{n \to \infty} \sum_{i=0}^{n} [I+2\ K_s A]^i = - \frac{1}{2\ K_s} A^{-1}$$

where the formula for the sum of a geometric series has been used. That is,

$$\sum_{i=0}^{\infty} (1 + 2\ K_s\ \lambda)^i = \frac{1}{1-(1+2\ K_s\lambda)} = \frac{-1}{2\ K_s\lambda}$$

Thus, in the limit, equation (2.24) becomes

$$\lim_{n \to \infty} E[W(n+1)] = Q^{-1}A^{-1}\ Q\ \ V_{dx} = V_{xx}^{-1}\ V_{dx}$$

which is the Wiener-Hopf solution in (2.17).

Convergence of $E[W(n+1)]$ to (2.17) is obtained if and only if the diagonal terms of $[I+2\ K_s A]$ all have magnitudes less than unity, and since all eigenvalues in A are positive (the auto-covariance matrix,

V_{xx}, is positive definite), the bounds on K_s are given by

$$| 1 + 2 K_s \lambda_m | < 1 \quad \text{or} \quad \frac{-1}{\lambda_m} < K_s < 0$$

where λ_m is the maximum eigenvalue of V_{xx}. The convergence condition can be related to the total input power as follows:

$$\lambda_m \leq \text{trace } [V_{xx}] = \sum_{i=1}^{N} E[x_i^2] = \text{Total input power.}$$

Therefore;

$$\frac{-1}{\sum_{i=1}^{N} E[x_i^2]} < K_s < 0.$$

For a slow, precise adaptation K_s is usually chosen such that

$$\frac{1}{\sum_{i=1}^{N} E[x_i^2]} >> |K_s|.$$

It is believed that the assumption of independent successive input vectors used for the convergence proof is overly restrictive. Griffiths [13] has shown that adaptation using highly correlated successive samples converges to the Wiener solution, but leads to higher steady-state mean-square error. Thus, we can conclude that for short-term stationary signals, with the feedback estimate, K_s, being bounded by (2.25), the

weight vector, after a large number of iterations, is expected to converge to the Weiner solution that best maps the input $x(n)$ to the desired response $d(n)$.

2.5 Adaptive LMSTDE Approach

The LMS adaptive filter described in the previous section has been widely applied in situations where the statistics of the inputs are either unknown or partially unknown. Some applications include noise cancelling [14], line enhancing [14], prediction [11], spectrum analysis [13], and adaptive array processing [15]. A recent application of the filter is time delay estimation by F. Reed, P. Feintuch and N. Bershad [16]. This application is demonstrated in Figure 2.3, where the adaptive filter has a slightly different structure. The filter to be considered here has only two inputs: a primary input $x(n)$ and a secondary input (desired response) $d(n)$. The primary input is fed to a tapped delay line to generate the adaptive filter input signals. In this case, the input vector at the nth iteration becomes

$$X^T(n) = [x_1(n) \quad x_2(n) \quad .. \quad x_n(n)]$$

$$= [x(n) \quad x(n-1) \quad ... \quad x(n-N+1)]$$

(for a filter of length N)

and the weights are updated with the arrival of each new data sample $x(n)$.

Let $x_1(t)$ and $x_2(t)$, in Figure 2.3, be given as in equation (2.1) with the same assumptions. Let us consider the discrete time version of (1) for the following analysis. That is,

$$x_1(n) = S_1(n) + m_1(n)$$

and $$x_2(n) = S_2'(n) + m_2(n) = \alpha\, S_1(n-D) + m_2(n)$$

where D is a positive integer less than N, representing the discrete time delay between $S_1(n)$ and $S_2(n)$.

The adaptive filter inputs are then given by

$$x(n) = x_1(n) = S_1(n) + m_1(n) \tag{2.26a}$$

and $$d(n) = x_2(n) = \alpha\, S_1(n-D) + m_2(n) \tag{2.26b}$$

The adaptive filter, with inputs given by (2.26), can be thought of as a system attempting to insert a delay equal to the propagation delay between the two sensors in the primary filter input, $x(n)$, aligning the signal component in time prior to subtraction to produce the error signal. Hence, one weight of the filter corresponding to the correct delay would be unity and all other weights zero. In practice, due to the fact that the filter must interpolate between its discrete taps to provide delays that are noninteger multiples of the sample time, the weights converge to a shape that is peaked at the correct delay.

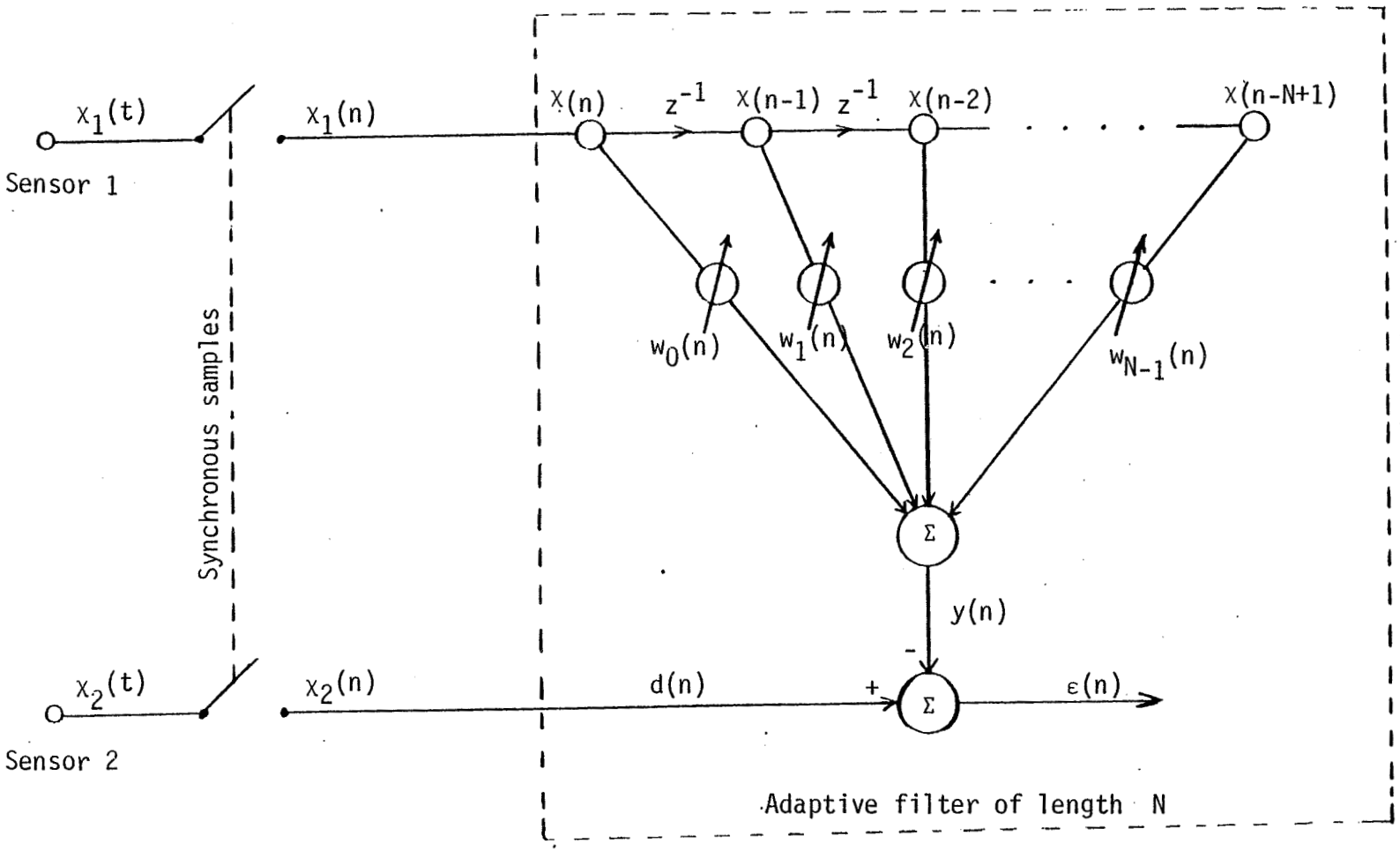


Figure 2.3. Modeling TDE using adaptive filter.

Therefore, determination of the delay, as in the GCC method, requires estimation of the peak of the adaptive filter weight vector.

To examine the shape of the weight vector, let us consider the frequency domain window corresponding to the least-mean-square set of weights, $W_{LMS} = (V_{xx})^{-1} V_{dx}$. That is,

$$W_{LMS}(f) = F[W_{LMS}(k)] = \frac{G_{dx}(f)}{G_{xx}(f)} .$$

For $d(n)$ and $x(n)$ given by (2.26), this becomes

$$W_{LMS}(f) = \frac{\alpha\, G_{S_1 S_1}(f)}{G_{S_1 S_1}(f) + G_{m_1 m_1}(f)}\, e^{-j2\pi fD} \qquad (2.27)$$

Converting back to time domain and using the convolution theorem, the kth weight at the nth iteration is

$$W_{LMS}(k) = F^{-1}\left[e^{-j2\pi fD}\right] \circledast F^{-1}\left[\frac{\alpha\, G_{S_1 S_1}(f)}{G_{S_1 S_1}(f) + G_{m_1 m_1}(f)}\right]$$

$$= F^{-1}\left[e^{-j2\pi fD}\right] \circledast \int_{\infty}^{\infty} \left[\frac{\alpha\, G_{S_1 S_1}(f)}{G_{S_1 S_1}(f) + G_{m_1 m_1}(f)}\right] e^{j2\pi fK} df \qquad (2.28)$$

where $0 \leq K \leq N-1$. Equation (2.28) is to be compared with equation (2.9), where $W_{LMS}(K)$ is a discrete time version of $R_{x_1 x_2}(\tau)$ obtained using the Roth processor. It is important at this point to note that

equations (2.27) and (2.28) do not depend on $G_{m_2 m_2}(f)$. Therefore, as suggested by Ahmed and Carter [17], it is desirable in this adaptive approach to have the signal with lower S/N ratio as the secondary input to the filter. From equation (2.27) we can see that when

$$G_{m_1 m_1}(f) = 0 \tag{2.29a}$$

or,

$$G_{m_1 m_1}(f) = \text{const.} \times G_{S_1 S_1}(f) \tag{2.29b}$$

we have $W_{LMS}(f) = \beta\, e^{-j2\pi fD}$, where β is a real constant. In this case, equation (2.28) becomes

$$W_{LMS}(K) = \beta \times F^{-1}\left[e^{-j2\pi fD}\right] \tag{2.30}$$

Let us examine equation (2.30) for two types of input signals. First, consider the case when the input signals are broad-band, i.e. $-\frac{f_s}{2} < f < \frac{f_s}{2}$, where f_s denotes the sampling rate. Evaluating (2.30) for this type of signal yields

$$W_{LMS}(K) = \beta \int_{-\infty}^{\infty} e^{-j2\pi fD} \times e^{j2\pi fK} df = \beta \int_{-f_{s/2}}^{f_{s/2}} e^{-j2\pi fD} \times e^{j2\pi fK} df$$

$$= \beta\ \text{sinc}\ \{\pi[K-D]\} \tag{2.31}$$

where $\text{sinc}(\cdot) = \frac{\sin(\cdot)}{(\cdot)}$ and $0 \leq K \leq N-1$. In this case the weight vector has the shape of a sinc function peaking with amplitude equal to one when $K-D = 0$ or $K = D$. The first zero crossing of (2.31) happens for

$$\pi(k-D) = \pm \pi \Rightarrow K = D \pm 1$$

Hence, the main lobe of the sinc function is only two resolutions wide, simulating a delta function. However, as in the Roth processor, when the condition (2.29) on $G_{m_1 m_1}(f)$ is not met, the main lobe of the sinc function will be spread out by convolving with the integral part of (2.28).

A second relevant case is one in which the input signals are band-limited in frequency. Evaluating equation (2.30) for band-limited signals leads to a slightly different form of the weight function as discussed by Ahmed and Carter [17]. Consider an ideal band-limited signal as follows

$$G_{S_1 S_1}(\omega) \neq 0 \qquad \text{for} \quad \omega_0 \leq |\omega| \leq \omega_1$$

$$G_{S_1 S_1}(\omega) = 0 \qquad , \qquad \text{elsewhere}$$

where the conversion to the radian frequency, ω, was made for the ease of analysis only. For this type of signal, equation (2.30) becomes

$$W_{LMS}(K) = \beta \int_{-\pi}^{\pi} e^{-j\omega D}\, e^{j\omega K}\, d\omega = 2\beta \int_{\omega_0}^{\omega_1} e^{+j\omega(K-D)} d\omega$$

$$= 4\beta\, \omega_b\, \text{sinc}\,[\omega_b(K-D)]\, \cos\,[\omega_c(K-D)] \qquad (2.32)$$

where $\omega_b = \frac{\omega_1 - \omega_0}{2}$ and $\omega_c = \frac{\omega_1 + \omega_0}{2}$

The peak in the weight function still occurs at the correct delay, that is, for $K-D = 0$ or when $K=D$, but the main lobe is spread out. That can be seen by examining the first zero crossing of equation (2.32). The first zero in the sinc function occurs for

$$\omega_b(K-D) = \pm\pi \rightarrow K = \frac{\pm\pi}{\omega_b} + D$$

while the first zero in the cosine function occurs for

$$\omega_c(K-d) = \pm\frac{\pi}{2} \rightarrow K = \frac{\pm\pi}{2\omega_c} + D$$

Since $2\omega_c \geq \omega_b$, the first zero in the cosine function occurs sooner than in the sinc and yields a main lobe width of $2 \times \frac{\pi}{2\omega_c} = \frac{\pi}{\omega_c}$. So for band-limited input signals, the spreading in the peak of the weight function is inversely proportional to the band width. For a white input signal, $\omega_c = \frac{\pi}{2}$ and the main lobe width is two resolutions as pointed out earlier.

Other issues that should be considered in this approach include interpolation, dynamic behavior of the filter, and the filter length. The time delay obtained from the estimated discrete weight function, $W_{LMS}(K)$, is an integer multiple of the sampling period while in practice the actual delay between the input signals may not be. To avoid the error introduced by the discrete delay estimate the weight function should be interpolated to obtain its continuous counterpart $W_{LMS}(n,K)$, that is,

$$W_{LMS}(n,K) = \sum_{m=-p}^{p} W_{MS}(m,K) \text{ sinc } [\pi(n-m)] \qquad (2.33)$$

Now, the desired noninteger sample of the estimated delay is given by the value of n at which $W_{LMS}(n,K)$ is maximum. The study of the filter dynamic behavior by P. Feintuch, N. Bershad, and F. Reed [18], has shown a potential to track linearly changing time delays by just observing the peak weight move through the adaptive filter tapped delay line. It was also shown that the location of the peak of the weights lags the true time delay by an amount that depends on the delay rate, the signal correlation function, and the adaptive filter time response. Finally, the filter length should be chosen carefully to insure fast and yet accurate convergence. The lower bound on the number of weights is depicted by the maximum propagation delay between sensors. From equation (2.32) we can see that as K increases the weight function $W_{LMS}(K)$ approaches zero. Thus, the filter length can be safely

truncated in practice to the lower bound, in order to reduce the computational time.

This section is concluded by a representative example illustrating the capability of the adaptive filter to detect the time delay after convergence of the weight vector. Let $x_1(n)$ be a stationary random process with a correlation function given by

$$R(n) = a^{|n|} \quad \text{with} \quad -1 < a < 1$$

Assume that $m_1(n) = m_2(n) = 0$ and $x_2(n) = x_1(n-D)$ where D is a positive integer given by $0 \leq D \leq N-1$. The adaptive filter inputs are then,

$$x(n) = x_1(n)$$

$$d(n) = x_1(n-D)$$

The input vector on the tapped-delay line is

$$X^T(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-N+1)]$$

and the cross-covariance vector, V_{dx}, is

$$V_{dx}^T = E[d(n)X^T(n)] = E[d(n)x(n) \quad d(n)x(n-1) \cdots d(n)\,x(n-N+1)]$$

$$X^T(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-N+1)]$$

and the cross-covariance vector, V_{dx}, would be

$$V_{dx}^T = E[d(n)X(n)] = E[d(n)x(n) \quad d(n)x(n-1) \cdots d(n)\,x(n-N+1)]$$

$$= [R(-D) \quad R(1-D) \quad \cdots \quad R(0) \cdots R(N-2-D) \quad R(N-1-D)]$$

$$= [a^D \quad a^{D-1} \quad \cdots \quad [\cdots 1 \cdots] \quad a \cdots a^{N-2-D} \quad a^{N-1-D}]$$

The auto-covariance matrix, V_{xx}, of $x(n)$ is

$$V_{xx} = E[X(n)X^T(n)] = \begin{bmatrix} R(0) & R(1) & \cdot & \cdot & R(N-1) \\ R(1) & R(0) & \cdot & \cdot & \\ \vdots & & & & \vdots \\ R(N-1) & & \cdot & \cdot & R(0) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & a & a^2 & \cdot & \cdot & a^{N-1} \\ a & 1 & a & \cdot & \cdot & \cdot \\ a^2 & & & & & \cdot \\ \vdots & & & & 1 & a \\ a^{N-1} & \cdot & \cdot & & a & 1 \end{bmatrix}$$

and

$$V_{xx}^{-1} = \begin{bmatrix} 1 & -a & 0 & 0 & \cdots & & 0 \\ -a & 1+a^2 & -a & & \cdot\cdot & & \\ 0 & -a & 1+a^2 & & & & \\ 0 & 0 & -a & & & \vdots & \vdots \\ \cdot & 0 & 0 & & & & \\ \cdot & \cdot & \cdot & & & & 0 \\ \cdot & \cdot & \cdot & & & & \\ & & \cdot & \cdot\cdot & 0 & -a\ 1+a^2 & -a \\ 0 & 0 & 0 & \cdot\cdot & 0 & -a & 1 \end{bmatrix} \frac{1}{1-a^2}$$

$$W_{LMS} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

D^{th} row, i.e. the weight vector peaked at the correct delay.

2.6 Summary

The generalized cross correlation approach was reviewed with more attention paid to the Roth processor. The adaptive filter and its application to TDE problems was analyzed in detail. It was shown that the adaptive LMSTDE method is equivalent to the Roth processor and that the adaptation process using the LMS algorithm, under stated conditions, ensures the convergence of the weight vector to the Wiener-Hopf solution that presents the best mapping between the two inputs of the filter. This mapping resulted in a weight vector peaking at the correct time delay with a peak resolution that is a function of the frequency band of the signals and their statistical characteristics. The advantages of the adaptive LMSTDE over the GCC method can be summarized as follows:

- simpler implementation, since no FFT computations are involved
- no prior knowledge of the statistics of the signal and noise are necessary
- ability to track moving sources

- less computational time.

The advantages over the nonadaptive approach (i.e. direct implementation of the Wiener-Hopf equation) are

- simpler implementation, since no matrix operations are involved
- the ability to track moving sources
- the filter length can be easily increased in order to decrease the bias in the estimate.

Chapter 3

SYSTEM DESIGN

3.1 Introduction

The theory behind time-delay estimation techniques has been reviewed in Chapter 2. This theory can now be used in the design process of the system. Based on the anticipated input signals and assumptions that can be made about the statistical characteristics, the choice of a suitable TDE technique is made in section 2. Section 3 then presents the implementation of this technique and relevant design issues in the specific application on hand. The system specifications and the way they are related, derived, or determined are the subject of section 4. Finally, the design process is summarized in section 5.

3.2 Design Assumptions

The choice of the TDE approach and the assumptions on which it is based are the subject of this section. Not much study has been dedicated to the high frequency band (1 to 20 Hz) of infrasounds in the past, especially for sounds related to meteorological events. Thus, little is known about the statistics of these infrasounds and assumptions have to be made with regard to recent investigator's predictions. These assumptions will be listed below with a relevant discussion about each of them.

a. Single Source. More than one meteorological event can take place at one time producing a multiple delay between sensor

outputs. The potential to separate delays and identify the different sources is feasible but it is outside the scope of this study. So the design process is restricted to a single source of sound waves.

b. Stationary Signal. The stationary assumption is safely made because the data file spans a time window that is only a fraction of the event's lifetime. The lifetime of a meteorological event can be anywhere from a few minutes, in the case of a microburst for example, to several hours in the case of a frontal system [1], [19]. This time frame is to be compared with the file time that, as will be seen later, is 20 seconds.

c. Uncorrelated Signal and Noise. Infrasounds emitted by a relatively remote source cannot be expected to correlate with local noise present at the microphone array.

d. Uncorrelated Noise at the Different Sensors. This assumption can be justified by having a long array foot so that local noise at one sensor is not swept across the array.

e. Frequency Band. The low frequency band (10^{-6} to about 1 Hz) of the infrasound spectrum is more related to static pressure fluctuations that are part of a turbulent motion. Since in this project, the intention is to detect acoustic (propogating wave) energy and as little as possible of the turbulent pressure field, the frequency band should be in the higher part of the spectrum. The 1-to-16-Hz band has been experimentally studied by Eric S. Posmetier [20]. He concluded that "the 1-to-16-Hz infrasound may be radiated by clear air turbulence

and may be a basis for a remote passive detection system." Based on the physics of infrasounds and Posmetier's prediction, a signal frequency band of 1-to-20-Hz is anticipated.

f. Spectral Shape of Signal and Noise. Based on past experience, the signal and noise spectra are expected to peak in the range .5-to-3 Hz.

Now that the assumptions about the data have been established, the choice of a suitable time-delay estimation approach can be made. The data statistical characteristics required for the Generalized Cross-Correlation method have been assumed in a, b, c, and d. Thus, the GCC method is applicable in this passive detection problem. As was shown in section 2.2, the basic cross-correlation has an undesirable spreading in its peak, which can be improved by a frequency weighting. The weighting proposed by Roth leads to a weighted correlation given by

$$R_{x_1 x_2}(\tau) = \delta(\tau-\tau_d) \circledast \int_{-\infty}^{\infty} \frac{\alpha G_{S_1 S_1}(f)}{G_{S_1 S_1}(f) + G_{m_1 m_1}(f)} e^{j2\pi f\tau} df$$

For the assumption made in f, about a similar spectral shape for signal and noise [i.e., $G_{mm}(f) \cong$ const. $G_{SS}(f)$], this correlation becomes

$$R_{x_1 x_2}(\tau) \cong \delta(\tau-\tau_d)$$

Thus, the Roth processor is appropriate for the assumed data, since its output is a delta function with a magnitude equal to one at the correct

delay. Implementing the Roth processor in the frequency domain, however, has the disadvantages discussed in section 2.3. On the other hand, the advantages discussed in section 2.6 make the choice of implementing this processor in the time domain, using the LMS adaptive algorithm, evident. The choice of the TDE approach has been made. The question now is whether the adaptive filter discussed in section 2.5 is totally suitable for this application. This is the subject of the next section.

3.3 Filter Design

The filter structure discussed by F. Reed, P. Feintuch, and N. Bershad [16], and presented in section 2.5, requires prior lead-lag information about the monitored signals in order to successfully estimate the delay. The delay estimate can be made only if the leading signal is fed as the primary input to the filter. This can be seen by examining Figure 2.3 where the input vector

$$X^T(n) = [x(n) \quad x(n-1) \quad x(n-2) \quad \cdots \quad x(n-N+1)]$$

component of the second sensor output, $d(n)$ must lag $x(n)$ so that the delay estimate can be made.

In acoustic detection problems, the lead-lag information is unknown. This fact results in the necessity of applying the algorithm twice to the same data, swapping channels the second time, in order to get the lead-lag information as well as the delay estimate. This difficulty can be solved by introducing an $\frac{N}{2}$ point delay in the

secondary filter input as illustrated in Figure 3.1. Introducing this delay in $d(n)$ leaves enough leading as well as lagging information in $X(n)$ to be compared with $d(n)$ to eventually produce the estimate.

In this implementation, the filter inputs will have a slightly different form than before. With discrete time signals given as in Chapter 2,

$$\chi_1(n) = S_1(n) + m_1(n)$$

$$\chi_2(n) = S_2(n) + m_2(n) = \alpha\, S_1(n-D) + m_2(n)$$

the filter inputs are now given by

$$\chi(n) = \chi_1(n) = S_1(n) + m_1(n) \tag{3.1a}$$

and

$$d(n) = \chi_2(n-N') = \alpha\, S_1(n-N'-D) + m_2(n-N') \tag{3.1b}$$

where $0 \leq n \leq$ record length, $|D| \leq N'$, N' = integer $(\frac{N}{2})$, and N is an odd positive integer equal to the filter length. If, as assumed earlier, $m_1(n)$ and $m_2(n)$ are uncorrelated, having $m_2(n)$ delayed by N' points have no bearing on the analysis. All the analysis, presented in Chapter 2, of the weight vector is still applicable here except that D must be replaced by $D+N'$ everywhere in the derivations. Equation (2.31) becomes in this case

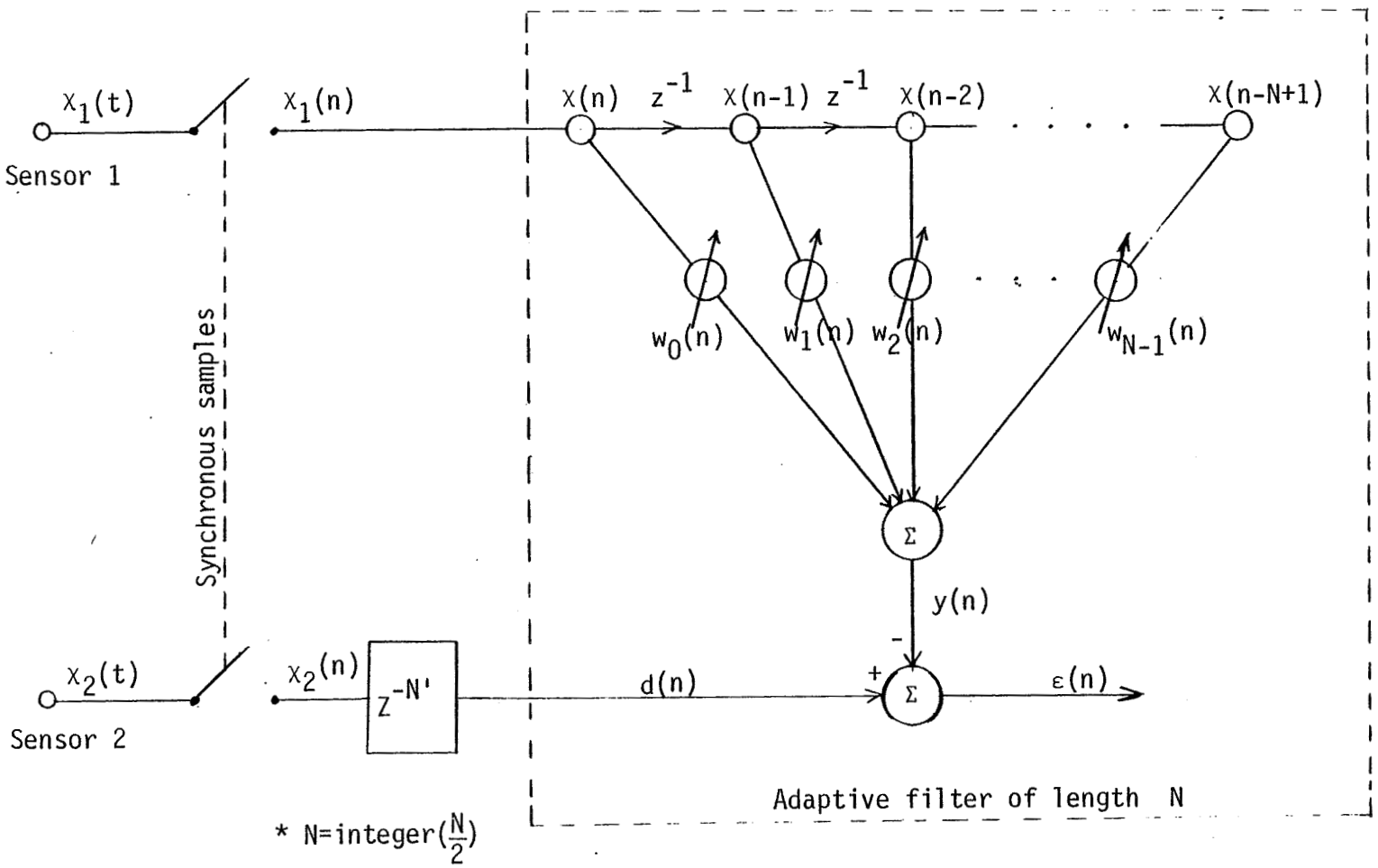


Figure 3.1 TDE lead-lag model.

$$W_{LMS}(K) = \beta \text{ sinc } \{\pi [K - (D+N)]\} \qquad (3.2)$$

As can be seen from (3.2), the weight vector now peaks at $K = D+N'$. So when $D=0$, the weights peak at N' (the middle weight), and the peak is shifted right or left with positive or negative values of the delay, respectively.

In section 2.4, it was shown that convergence of the weight vector to the Weiner-Hopf solution can be obtained only if the stability constant K_s is bounded by equation (2.25), that is,

$$\frac{-1}{\text{Input power}} = \frac{-1}{\sum_{i=1}^{N} E[x_i^2]} < K_s < 0$$

However, maintaining a constant K_s through the adaptation process can lead to instability in situations where a rapid change in the power level of the signal may occur. For this reason, K_s is updated in this design with every iteration of the algorithm. An estimate of the input power of the signal, based on the signal components in the tapped delay line, is made with the arrival of each data sample. The value $|K_s|$ is then chosen to be less than the inverse of the power estimate by a factor to be experimentally determined. Updating K_s in this fashion ensured a fast, stable convergence of the weight vector regardless of the power level of the input.

The final design issue is the convergence criteria. Due to the fact that our sampled data records may not contain any coherent signals

all the time, and due to the fact that the shape of the weight vector is unpredictable in these situations, a convergence criterion must be derived to distinguish between coherent and noncoherent input data before attempting to estimate the delay. This criterion is proposed as follows. Let us monitor the peak of the weight vector at every iteration as the data record is processed through the algorithm. By storing the abscissa value (corresponding to the discrete time delay) of the peak with every iteration, a data sequence is produced that can be plotted to produce a TDE vs. time plot. A variance criterion can then be applied to this TDE vs. time plot to determine whether a convergence has occurred or not. The data sequence representing the abscissa values of the peaks is as long as the data record itself. Thus, for short-term stationary data, the variance criterion should be applied to a sliding window, of length less than the data records, through the sequence instead of computing the variance for the whole sequence at once. Whenever the variance criterion is met, the delay estimate is made based on the abscissa values contained in the window where the convergence has occurred. This provides a better estimate of the delay, compared to taking the abscissa value of the peak in the last iteration, especially in noisy environments.

A detailed description of our implementation of the adaptive LMSTDE algorithm is presented next in Figure 3.2, Figure 3.3, and Figure 3.4, by means of flowcharts. In Figure 3.2, it is assumed that the sampled data records have been stored in arrays x and d of length L, where L is the data record length. W is an N point array, where N is the filter length, storing the weight vector components. The abscissa

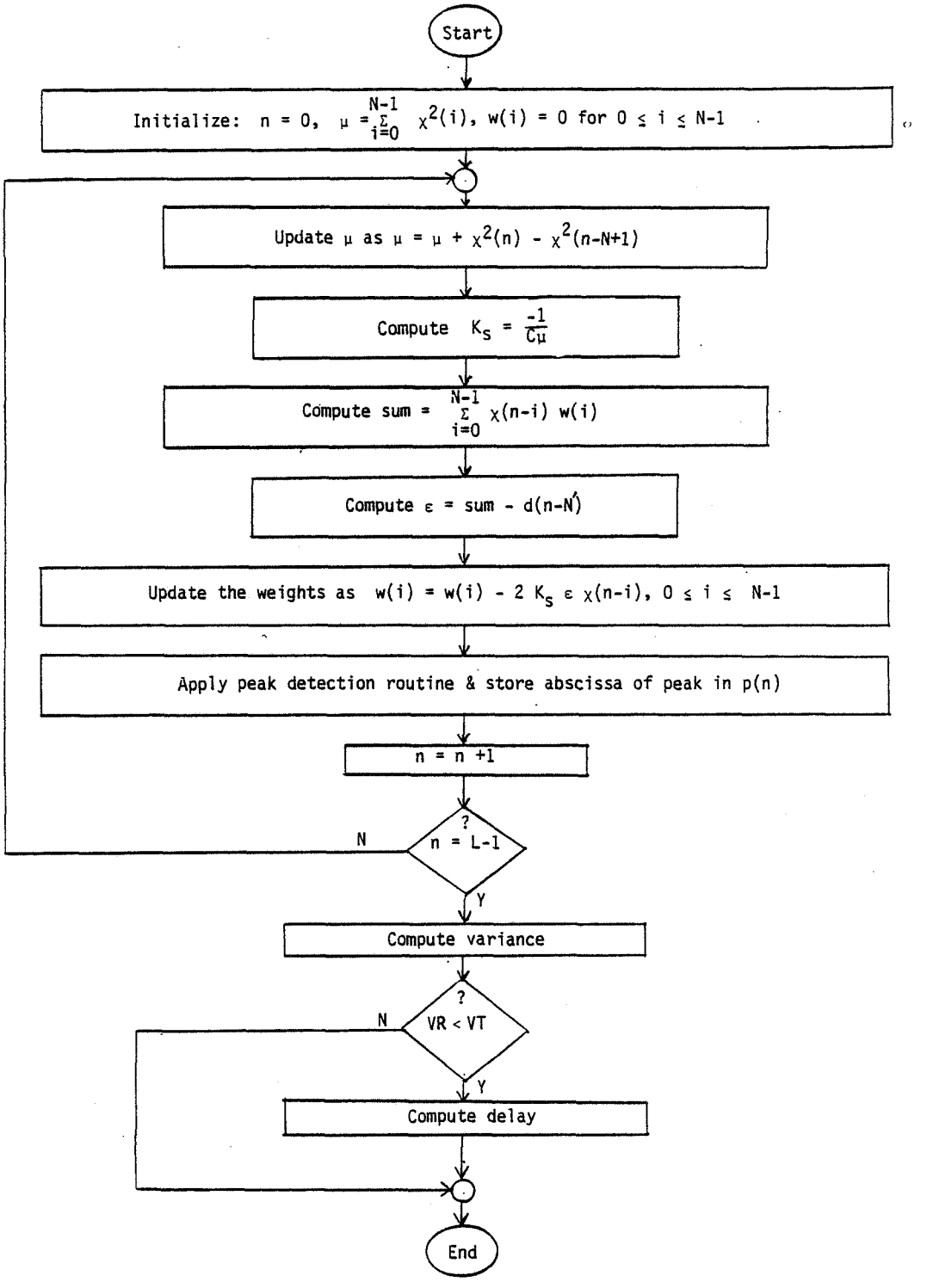


Figure 3.2. LMSTDE algorithm's flowchart.

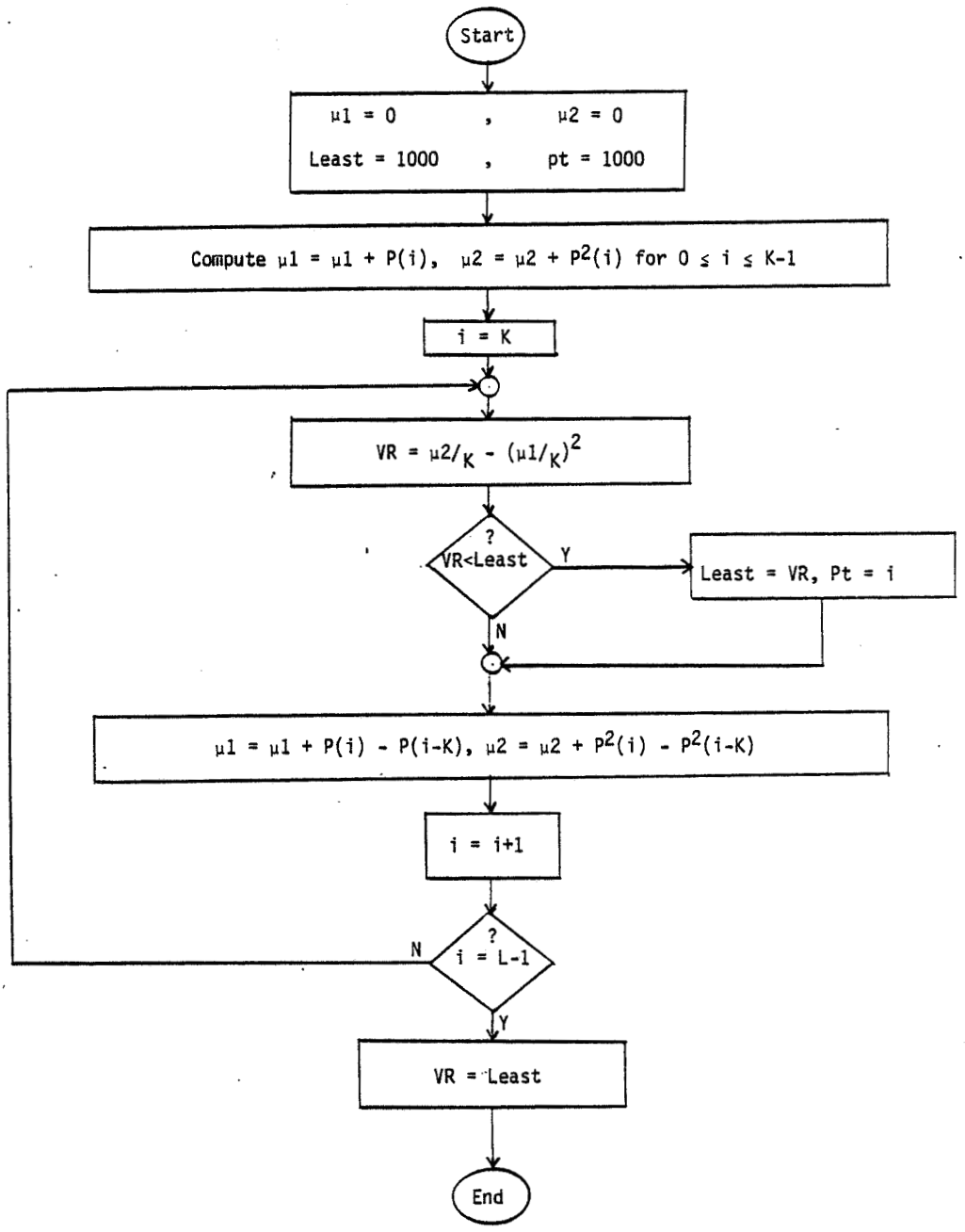


Figure 3.3. Variance computation.

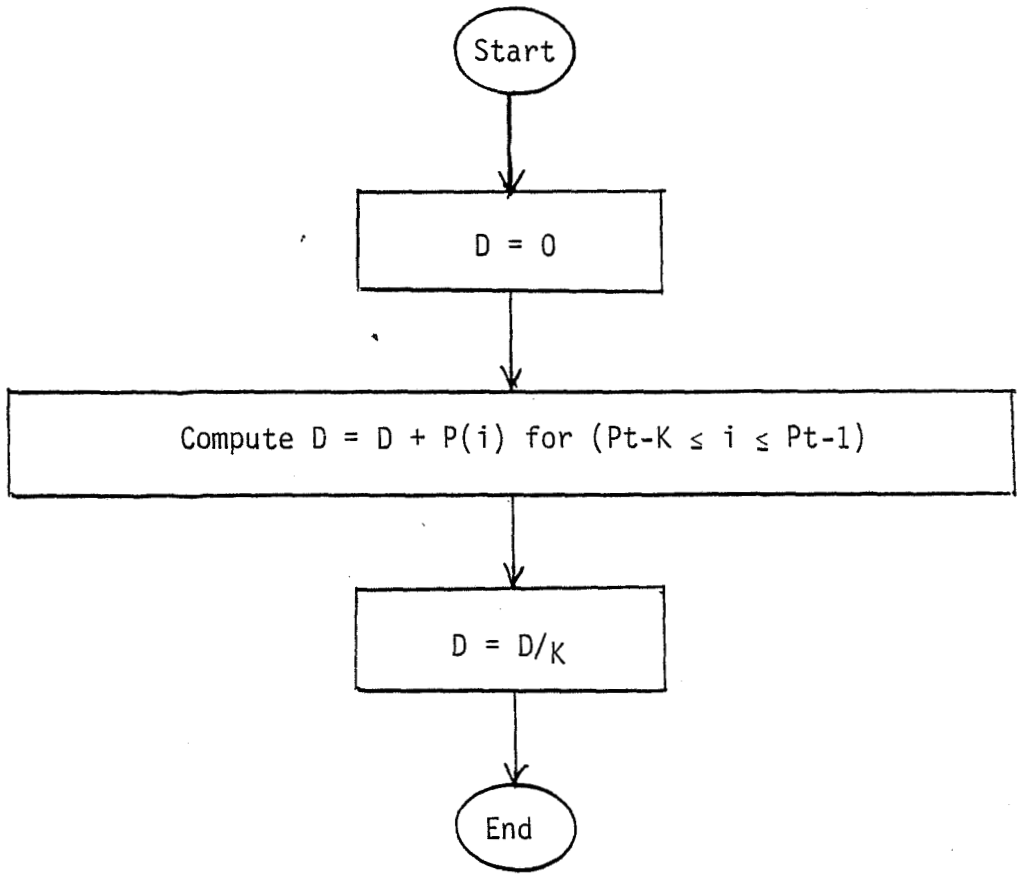


Figure 3.4. Delay estimate computation.

values of the peaks will be stored in an L point array P. VR and VT denotes the variance and the variance threshold, respectively. The estimate of the stability constant, K_s, is computed with every iteration as

$$K_s = \frac{-1}{C\mu} \tag{3.3}$$

where C is a predetermined factor and

$$\mu = \sum_{i=0}^{N-1} E[x_i^2(n)] = N \times E[x^2(n)] = \sum_{i=0}^{N-1} x^2(n-i)$$

The peak detection block in Figure 3.2 is a simple search routine, while the variance computation is described in Figure 3.3 and the delay computation in Figure 3.4. The variance computation is made, as described earlier, over a sliding window of length K, where $K < L$. The equation used for the variance estimate is

$$VR = \frac{1}{K} \sum_{i=0}^{K-1} P^2(i) - [\frac{1}{K} \sum_{i=0}^{K-1} P(i)]^2 \tag{3.4}$$

Finally, the delay estimate is made by simply averaging the delay values contained in the window that yielded the minimum variance level.

By inspecting the algorithm flowcharts, it is easily seen that the computational complexity is proportional to L x N. In practice L is the minimum number of iterations necessary for the convergence of the weight vector to the Wiener-Hopf solution. Therefore, for a specific

statistical type of data, L is a constant and the complexity becomes of order N. This design, accommodating positive and negative delays, implies a lower-bound on N of twice the maximum propagation delay between two sensors. Compared to the filter discussed in section 2.5, the filter length now is twice what it was before. With the complexity being of order N, it appears as if no speed improvement has been established over the previous design, where two passes were needed to obtain the delay estimate. Actual implementations, however, proved about 10% time saving. The time saving was mainly obtained by avoiding the overhead computation involved in every pass as can be seen in the flowcharts.

3.4 Specifications and Hardware

This section presents and justifies the specific parameter values in the algorithm implementation as well as in the hardware system used. These specifications are summarized in Table 3.1. They can be divided into three categories. The first category includes parameter values derived from the assumption made in the previous section about the signal frequency band. The second consists of parameters that were experimentally determined, while the ones related to hardware make up the third category.

The starting point in the first set of parameters is the frequency band that was specified as 1 to 20 Hz. An immediate result of this band is the sampling rate. A sampling rate of 40 Hz, satisfying the Nyquist rate, will not be adequate, the reason being that the signal has to be

Table 3.1. Parameter Specifications

PARAMETER	NOTATION	VALUE
Frequency Band	—	2 Hz - 20 Hz
Sample Rate	f_s	51.2 Hz
Array Foot	ℓ	800 ft.
A/D	—	12 bits
Quant. Level	—	2.4 mV/bit
File Length	M	1024 points
Record Length	L	128 points
Filter Length	N	77 points
Scale of Feedback Estimate	C	10
Variance Threshold	VT	1
Variance Window	K	48

low-pass filtered at 20 Hz with a non-ideal roll-off filter. To eliminate any alias effects produced by attenuated frequencies above 20 Hz, a sampling rate of 51.2 Hz was chosen. The exact value of 51.2 Hz is desirable because it produces a rounded frequency resolution of .2 Hz on a 256-point FFT spectrum. Given the sampling rate, the array foot length (distance between two sensors) becomes a compromise between computational speed and bearing resolution. The shorter the array foot, the less the (maximum) delay, and the lower the resolution. The filter length, and hence the speed is directly proportional to the maximum propagation delay as was seen in the previous section. The bearing

resolution is related to the array foot as

$$\Delta\theta = 90°-\cos^{-1}\left(\frac{\Delta t \times \text{speed of sound}}{\text{array foot length}}\right) \qquad (3.5)$$

where $\Delta\theta$ is the bearing resolution and Δt is the sampling period. To develop insight for the range of values involved, we can compute the filter length and bearing resolution for three different values of the array foot. ($\Delta t = 1/51.2$ sec, sound speed = 1080 ft/sec, filter length = $2\ D_{max}+1$)

Array foot	2000 ft.	800 ft.	100 ft.
Filter length	200	77	10
Bearing resolution	.6°	1.5°	12°

The first set of values implies a good resolution at the expense of slow speed while the third set implies a high speed computation at the expense of poor bearing resolution. The compromise is obtained in the middle set. Note that the chosen distance between sensors is not likely to violate our assumption of uncorrelated noise across the array. This is true since infrasound noise produced mostly by local wind fluctuations and small eddies are not expected to propagate over an 800 foot distance.

Experimentally determined parameters are the record length (number of iterations), scale of stability constant, variance window, and variance threshold. The first two parameters, and the second two, are related. The record length, L, should be the minimum number of iterations that leads to convergence in the weight vector to the Wiener-Hopf

solution. The rate of convergence is controlled by the stability constant, K_s, as was seen in equation (2.21) in section 2.4. Keeping one of the two constant and varying the other, the system performance was observed and the values were chosen as

$$L = 128 \text{ iterations}$$

$$C = \text{scale of } K_s = 10$$

The system performance, as L and C vary, is simulated in section 4.2. The variance window, K, and the variance threshold, VT, were used in the convergence criteria to determine whether the data is coherent or not. Ideal response would have a zero variance across a 128-point (record length) window, but obviously that is too tight a criterion in practice. Based on the response of the algorithm to some experimental data gathered, including space-shuttle launch noise, moderate values of 48 points and 1 were chosen for K and VT, respectively. It is worth mentioning at this point, that although the algorithm computation is based on 128 iterations, the data files sampled are 1024 points each. This file length is necessary for subsequent spectral analysis.

A block diagram of the system hardware and most of its specifications is presented in Figure 3.5. The type of sensors used in the array is the GLOBE 100 C Capacitor Microphone by GUS Manufacturing Inc. The high sensitivity (.2 volt/dyne/cm^2) and the low frequency response (.1 to (.1 to 500 Hz) on these Microphones make them very adequate for this application. Their dynamic range is 74 dB, .004 to 20 volts peak-to-peak

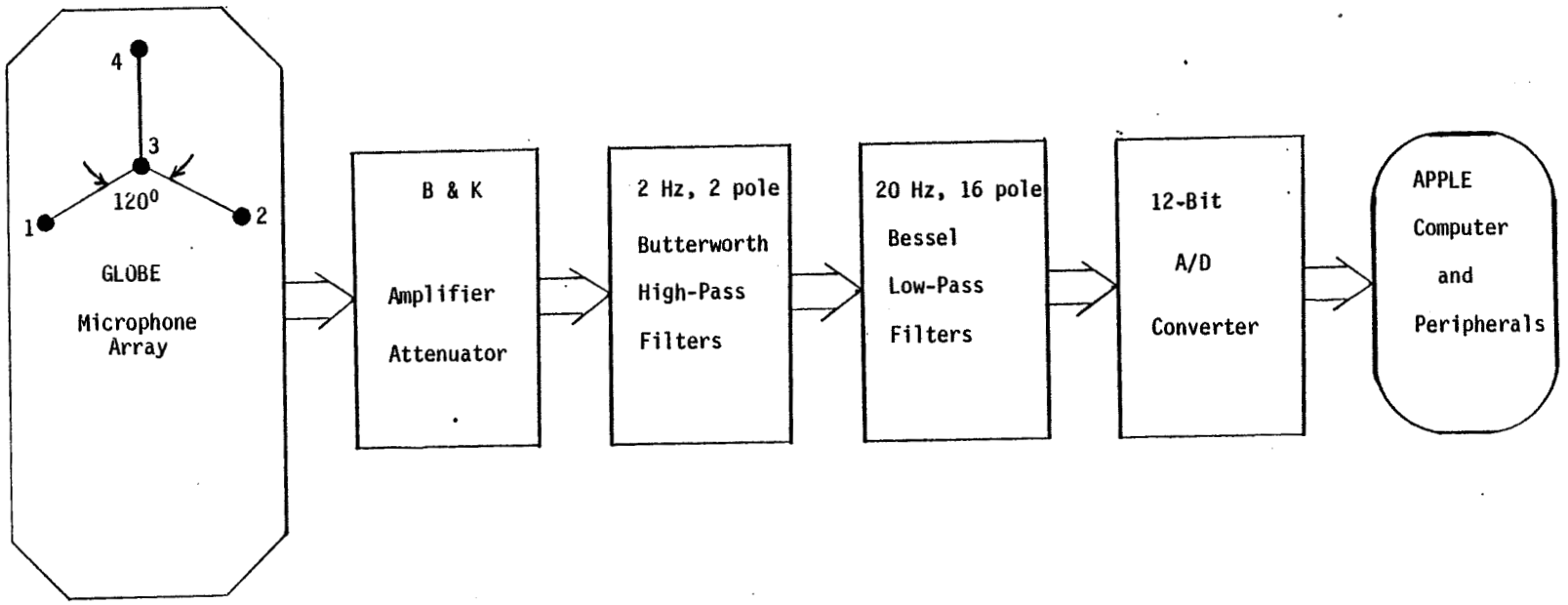


Figure 3.5. System block diagram.

peak. This dynamic range implies an electrical noise amplitude of (upto) 4 m volts p-to-p. To avoid any electrical noise in the sampled data, the analog-to-digital converter should have a quantization level of at least 2 mV/bit when no amplification is used. But since a large dynamic range A/D is obviously desirable, a 12-bit, 2.4 mV/bit A/D converter was chosen. An analog input range of -5V to +5V corresponds to a digital output range of 0 to 4095. Therefore, as long as no amplification is used on a microphone's output, the data are free from electrical noise. The specifications on the remaining hardware blocks are totally independent. The array configuration was intended to provide symmetry between the microphones so that delay-to-angle conversion can be made easier. Due to the experimental nature of this project and due to the lack of knowledge about the sound pressure level of infrasounds generated by meteorological events, the amplifier/attenuator block was necessary. Even though a signal frequency band of 1 to 20 Hz is anticipated, the input is high-pass filtered at 2 Hz in order to eliminate the strong wind noise in the region of .1 to 2 Hz. Finally, the choice of the APPLE computer, as the system processing unit, was made to meet the economical, portable system requirement.

3.5 Summary

The design process started by evaluating the anticipated signal and noise. The main assumption made was that of similar spectral shape for signal and noise over a frequency band of 1-to-20-Hz. This assumption led to the choice of the Roth processor which, in this case, produced a sharp peak at the correct delay in the weighted correlation function.

Examining the facts of section 2.6, proved that implementing the Roth processor in the time domain (rather than frequency domain as in the GCC method) using the LMS adaptive filter is advantageous. The specific filter design for this application was then presented. An $\frac{N}{2}$ -point delay in the secondary filter input was necessary to accommodate for positive and negative delays. Updating the stability constant, K_s, with every iteration was made for fast stable convergence of the weight vector when the power level of the signal may change rapidly. To discriminate against non-coherent input signals in the automatic detection mode of the algorithm, a convergence criterion was devised. This criterion was based on the variance level of the delay estimate with time. The design phase ended by schematically presenting the exact implementation of the design and specifying the software, as well as the hardware, parameters values.

Chapter 4

SIMULATION AND RESULTS

4.1 Introduction

Aided by the theory of TDE techniques reviewed in Chapter 2, the system design has been described in the previous chapter. This chapter presents an evaluation of the foregoing design. The response of the TDE algorithm, as presented in section 3.3 is evaluated in section 2. Its response for deterministic data sequences is simulated as the different parameters are individually varied. Its effectiveness in tracking time delays between real, long-range, infrasounds is also illustrated. Section 3 describes the complete software package, with the TDE algorithm being its central processing element, used in the actual field operation of the system. Its routine operation and general results over the period of June to September 1984, are also discussed. The system is evaluated with respect to man-made and weather-related infrasounds. The man-made infrasounds are discussed in section 4, with an illustrative example using signals from jet-powered aircrafts, while the weather-related infrasounds are the subject of section 5.

4.2 System Behavior

This section is dedicated to the study of the system response for deterministic input data as well as real infrasound waves generated by a far-field source. Deterministic data are used to study the system

behavior as the different parameters, with effect on the response, are varied one at a time. The type of data used in the simulation and the response for a nominal set of parameters are discussed in sub-section 1. The set of parameters can be divided into two groups. The first group consists of the adaptive filter parameters that were experimentally determined in the design. This group is the subject of subsection 2. The second group, containing the signal parameters (signal characteristics), is discussed in subsection 3. Finally the response for real signals consisting of the space shuttle launch noise is illustrated in the fourth subsection.

4.2.1 Nominal Response

The data used in this simulation are white data sequences generated on an Apple II computer using a random number generator function. White data are used for both signal and noise. Even though data with decaying spectra seem more appropriate for the simulation, white sequences were chosen for the ease in which they are generated. As explained in section 2.5, both types of data are expected to produce the same response since the weight vector, for signal and noise having the same spectral shape, is a sinc function regardless of what the spectral shape is. A 256-point sequence and its corresponding auto-power spectrum, representing the type of white sequences used, are plotted in Figure 4.1. The auto-power estimate is obtained by a 256-point FFT, and represented by the first 129 components since the last 127 are a mirror image of the first. The spectrum is expected to be completely white, in contrast to the one in the figure, as the number of overaged 256-point periodograms

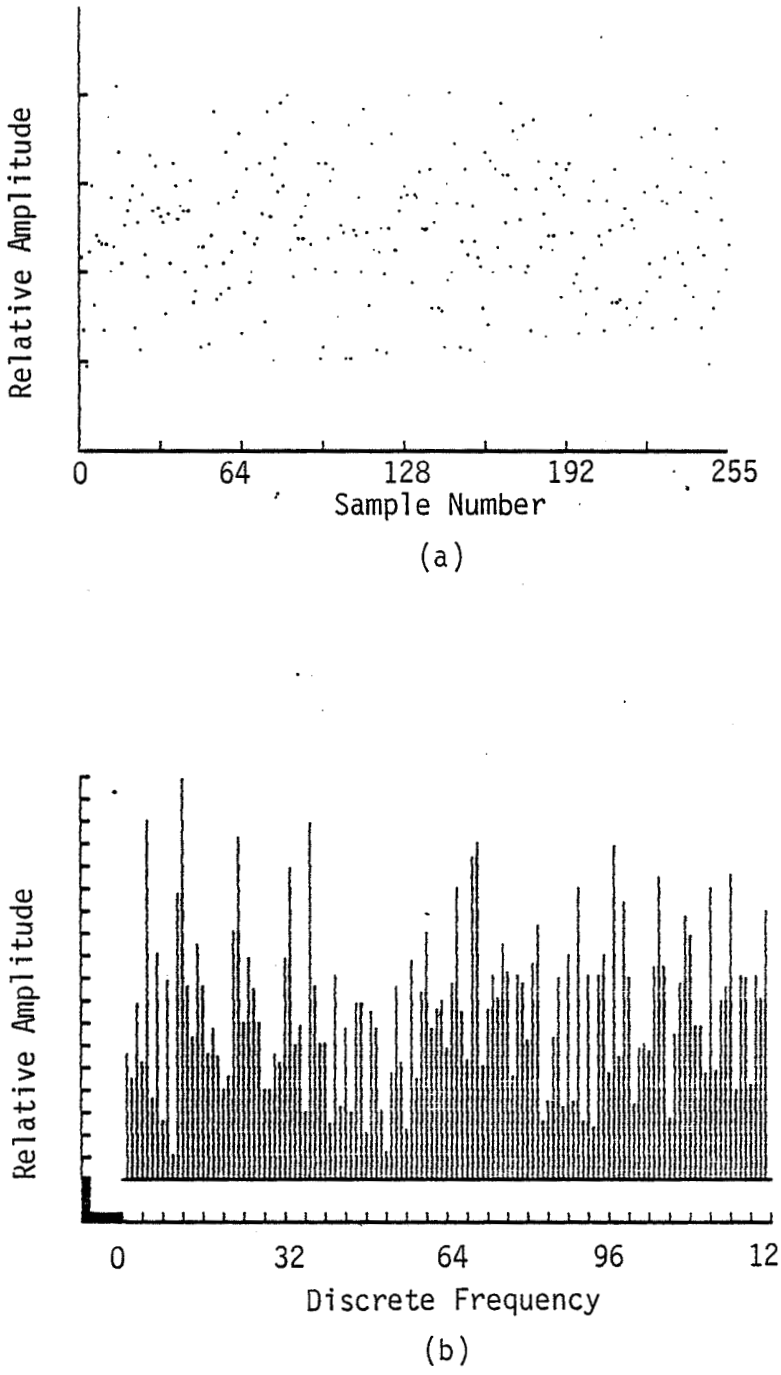


Figure 4.1. Type of random data sequences used for simulation. (a) 256-point sequence. (b) Its auto-power spectrum.

approaches infinity.

The nominal response of the algorithm for positive and negative delays is presented in Figure 4.2 and Figure 4.3, respectively. The positive delay in Figure 4.2 was generated in the input sequences as follows:

$$\chi(n) = R_1(n) + R_2(n) \tag{4.1a}$$

$$d(n) = R_1(n+4) + R_3(n) \tag{4.1b}$$

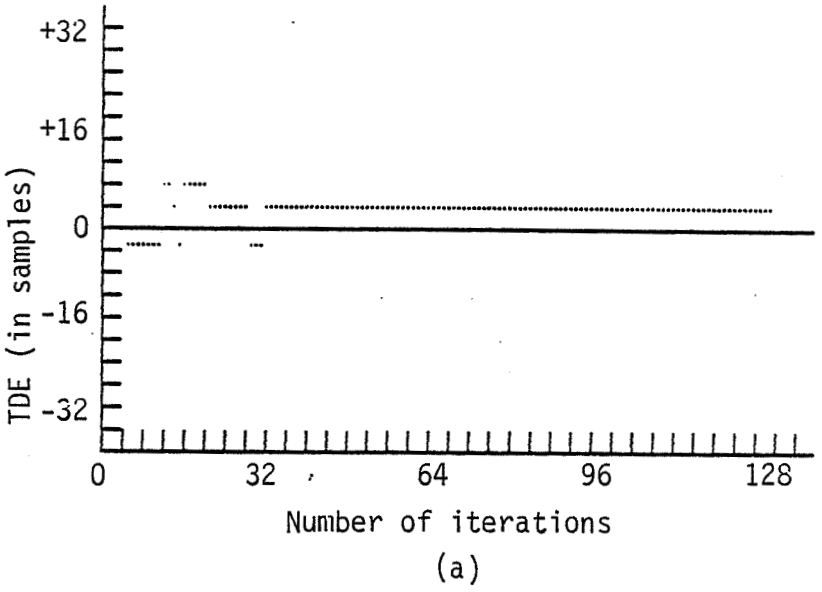
where $\chi(n)$ is the primary input to the filter, $d(n)$ is the secondary input, $R_1(n)$, $R_2(n)$ and $R_3(n)$ are uncorrelated random sequences. The embedded delay is four samples while the other parameters are set to their nominal values as specified in the figures. The negative delay was obtained by simply swapping the adaptive filter inputs, i.e.:

$$\chi(n) = R_1(n+4) + R_3(n)$$

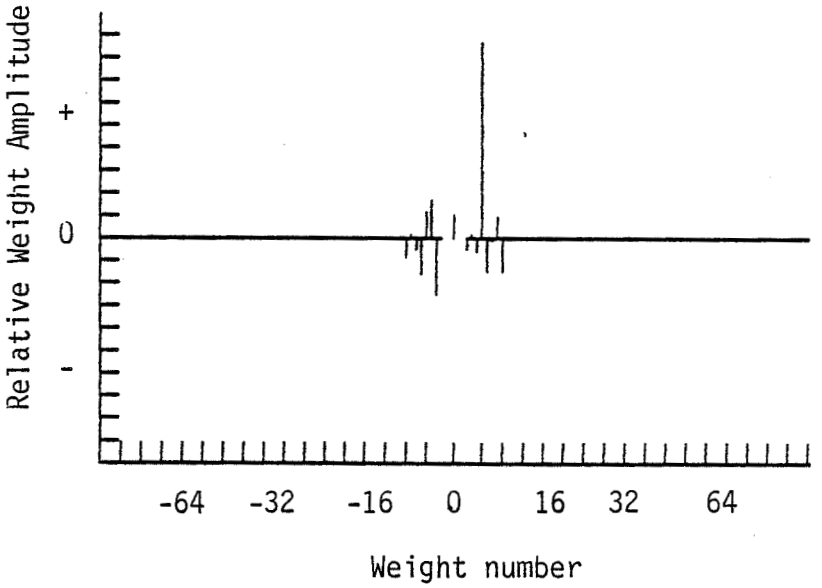
$$d(n) = R_1(n) + R_2(n)$$

The fact that $\chi(n)$ now lags $d(n)$ by four samples leads to a negative delay in the output of the filter as illustrated in Figure 4.3.

The first plot in both figures was obtained by monitoring the abscissa value of the peak in the weight vector at every iteration of the algorithm. The second plot is the final set of weights. The

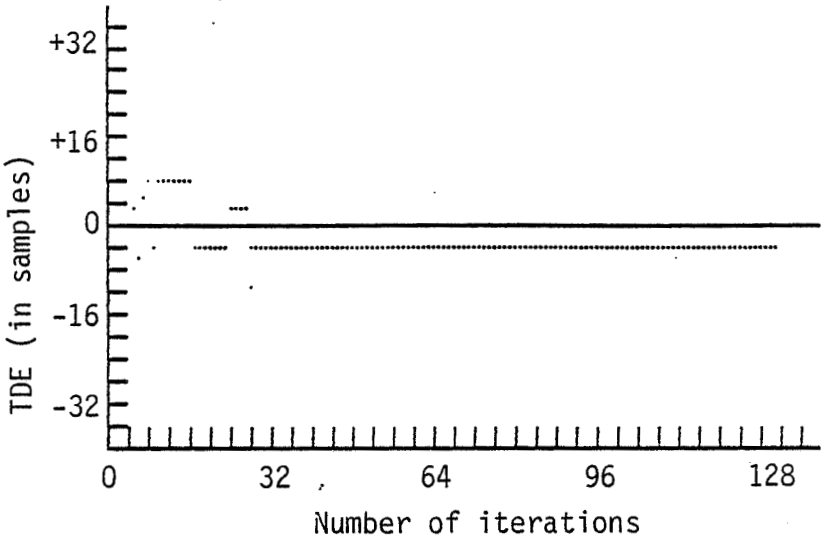


(a)

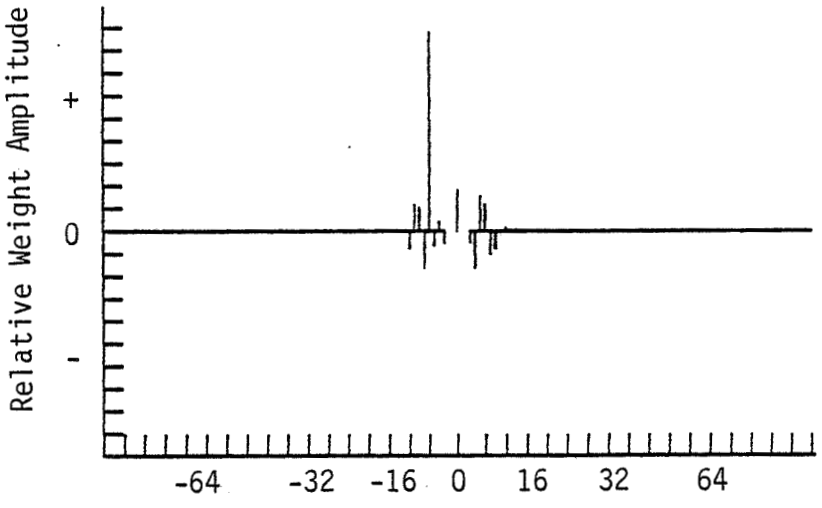


(b)

Figure 4.2. Nominal filter response for D = +4 samples. (C=10, N = 17, L = 128, SNR = 1). (a) TDE vs. number of iterations. (b) Final weight vector.



(a)



(b)

Figure 4.3. Nominal filter response for D = -4 samples. (C=10, N=17, L=128, SNR=1). (a) TDE vs. number of iterations. (b) Final weight vector.

capability of tracking positive and negative delays is evident from the figures. The "steady-state" delay has shifted from +4 to -4 in the TDE vs. time plot, while the peak in the weight vector has shifted right and then left by 4 units as expected. These plots are to be compared with the response obtained in the next two subsections where the various parameters are varied.

4.2.2 Filter Parameters

The filter parameters include the filter length (N), stability constant scale factor (C), and the record length (L). In the design process, a lower bound on N was set while C and L were experimentally determined. This simulation verifies the necessity of the lower bound on N and explores the basis on which C and L were determined. The input sequences used in this subsection are the ones previously given in section (4.1).

The lower bound on the filter length (number of weights) was derived in section 3.3 as $(2xD_{max}+1)$. Twice the maximum delay was necessary since the filter accommodates positive and negative delays. A number of weights less than the lower bound will obviously give no estimate. A very large number of weights, on the other hand, will not improve the response since the sinc function (the shape of the weight vector) approaches zero as the number of weights is increased. This behavior is exemplified for D=4 in Figure 4.4 with N=7 and in Figure 4.5 with N=33. The remaining parameters are set to their nominal values in both figures. The response in both cases is to be compared

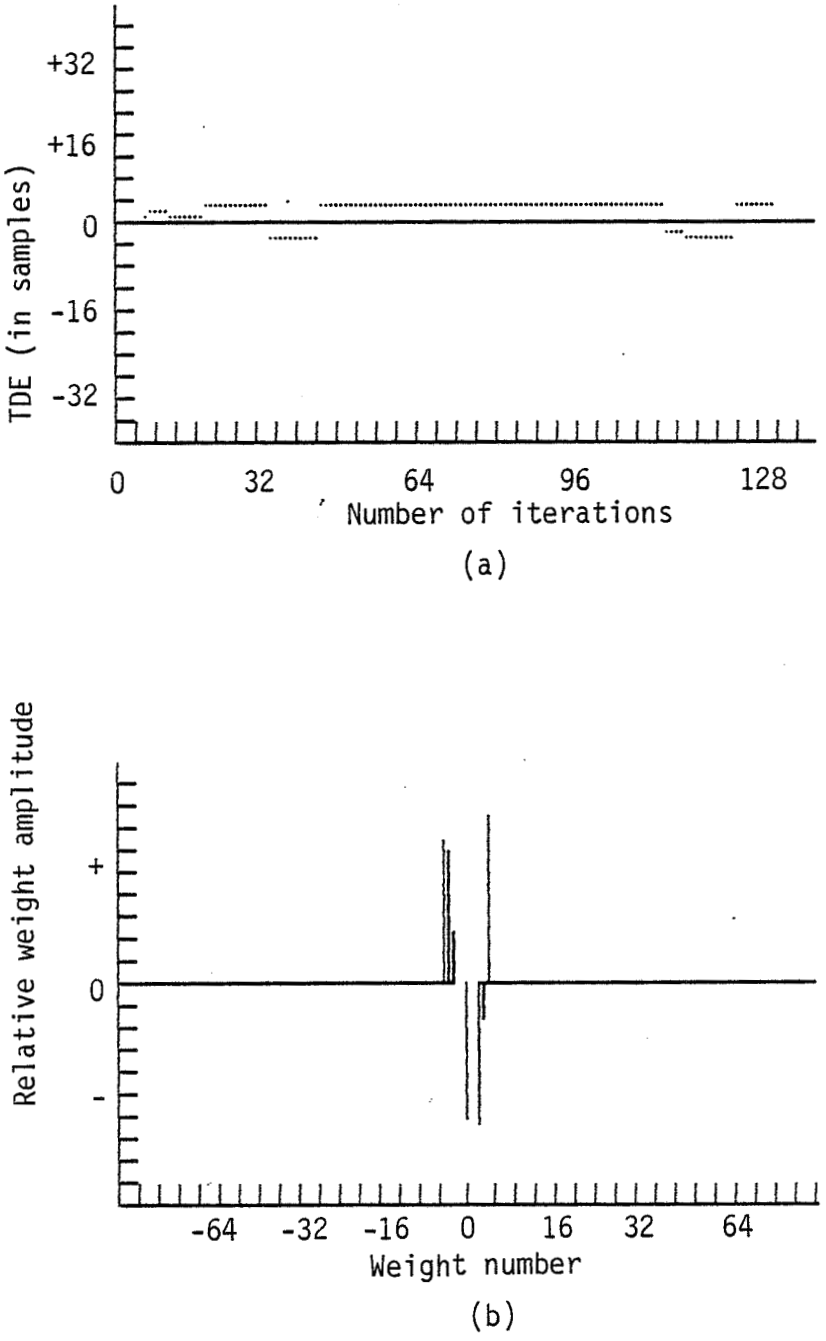


Figure 4.4. N=7. Delay can not be estimated with $N<2xD_{max}+1$. (D=4, C=10, L=128, SNR=1). (a) TDE vs. number of iterations. (b) Final weight vector.

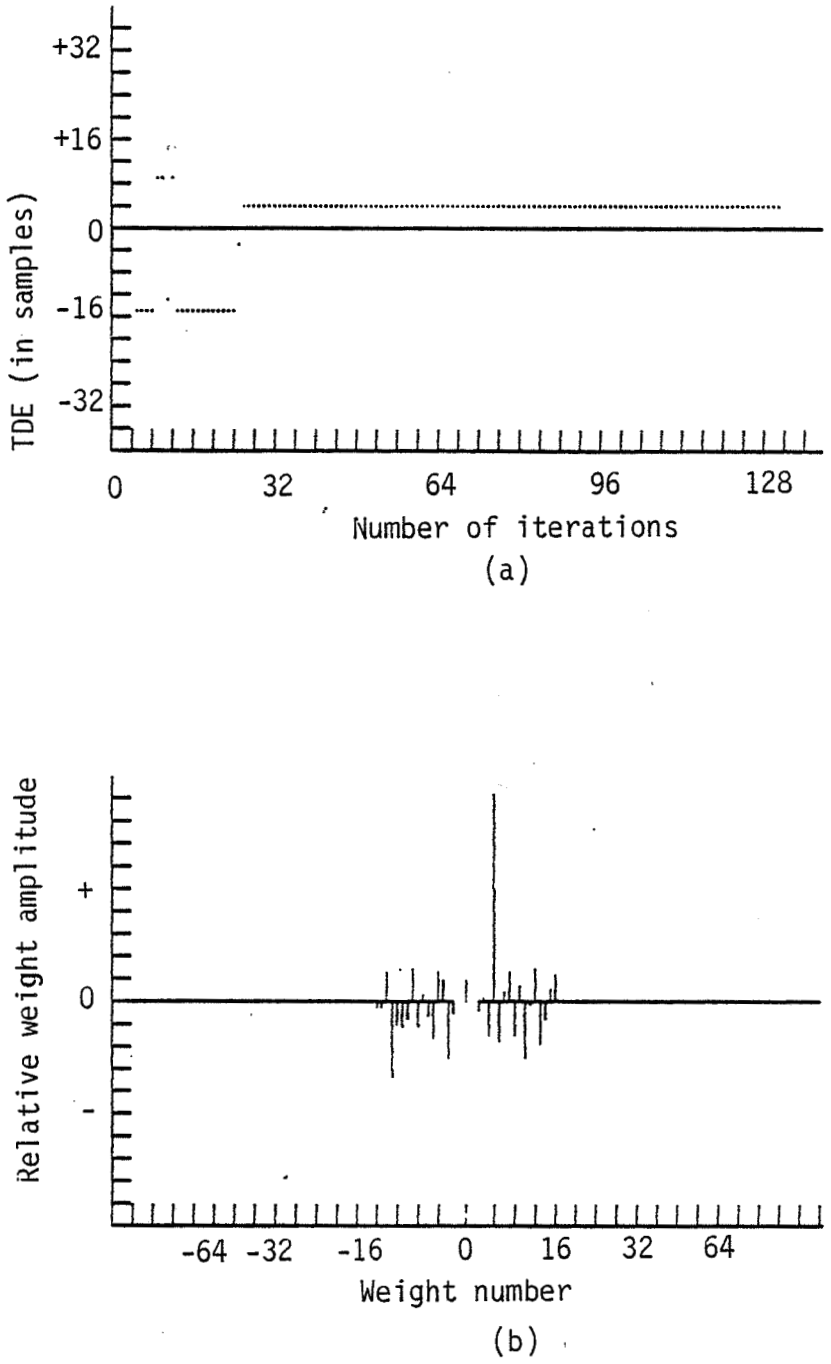


Figure 4.5. N=33. No improvement over N=17. (D=4, C=10, L=128, SNR=1). (a) TDE vs. number of iterations. (b) Final weight vector.

with the nominal response in Figure 4.2 where N=17.

The effect of varying C on the stability of the system is demonstrated in the next two figures. It was seen in section 3.3, the bounds on K_s are given by:

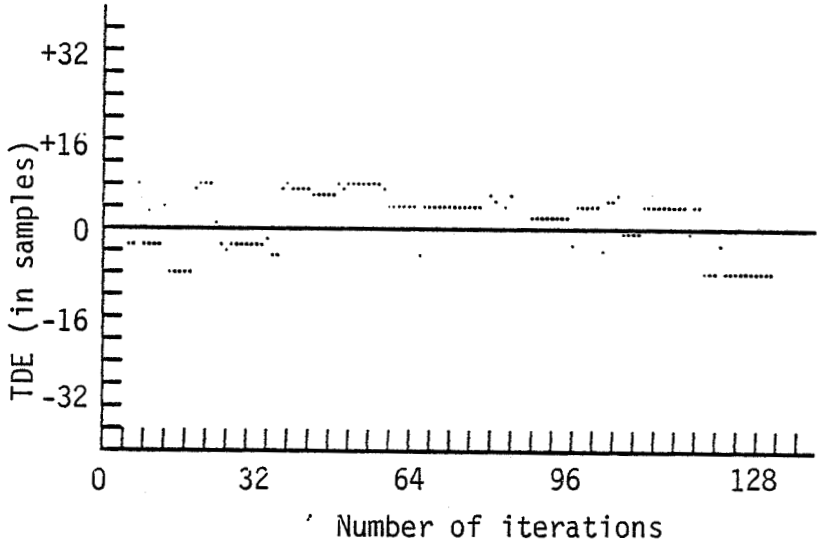
$$\frac{-1}{\sum_{i=0}^{N-1} \chi^2(i)} < K_s < 0$$

The value used in this implementation is given as

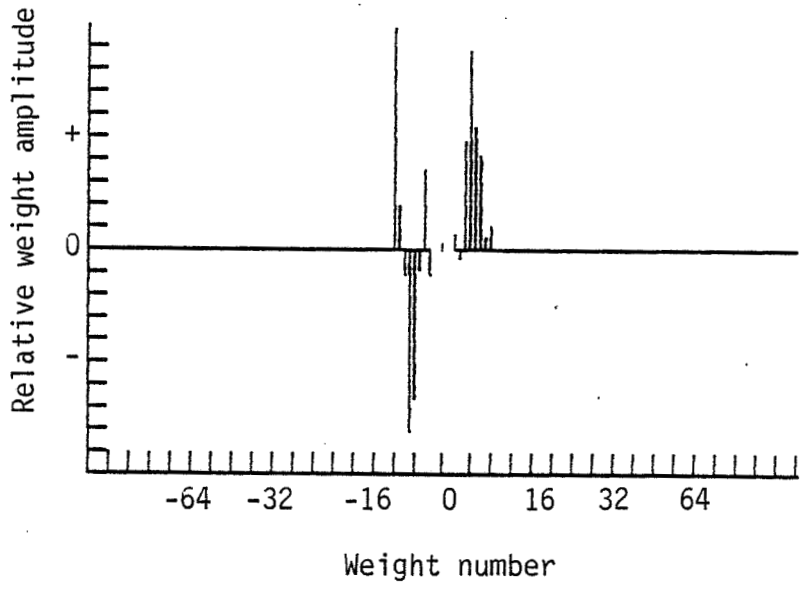
$$K_s = \frac{-1}{C \sum_{i=0}^{N-1} \chi^2(i)} = \frac{-1}{C\mu}$$

where C is a positive scale factor that has to be greater than one to satisfy the stability condition. The output of the filter is plotted in Figure 4.6 and Figure 4.7 for C=.5 and C=200, respectively. Having C less than one led to an unstable system as predicted by Widrow [11], while increasing its value behind ten hardly affected the response.

The record length (number of iterations) was chosen as 128. The purpose of this simulation is to show that a 128-iteration process is sufficient for the convergence of the weight vector to the Wiener-Hopf solution. This is illustrated in the next two figures in comparison with Figure 4.2. Figure 4.8 illustrates the ambiguity in the filter output caused by too small a number of iterations. The expected

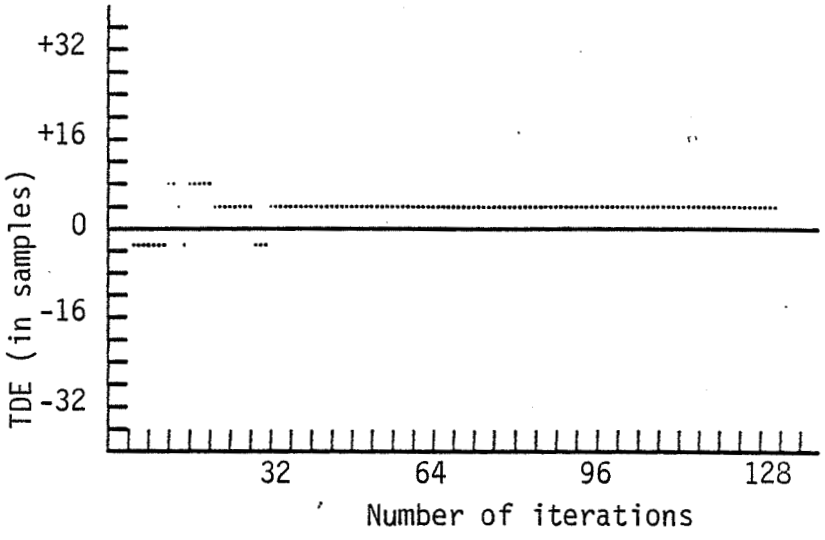


(a)



(b)

Figure 4.6. Unstability for C=0.5. (D=4, N=17, L=128, SNR=1). (a) TDE vs. number of iterations. (b) Final weight vector.



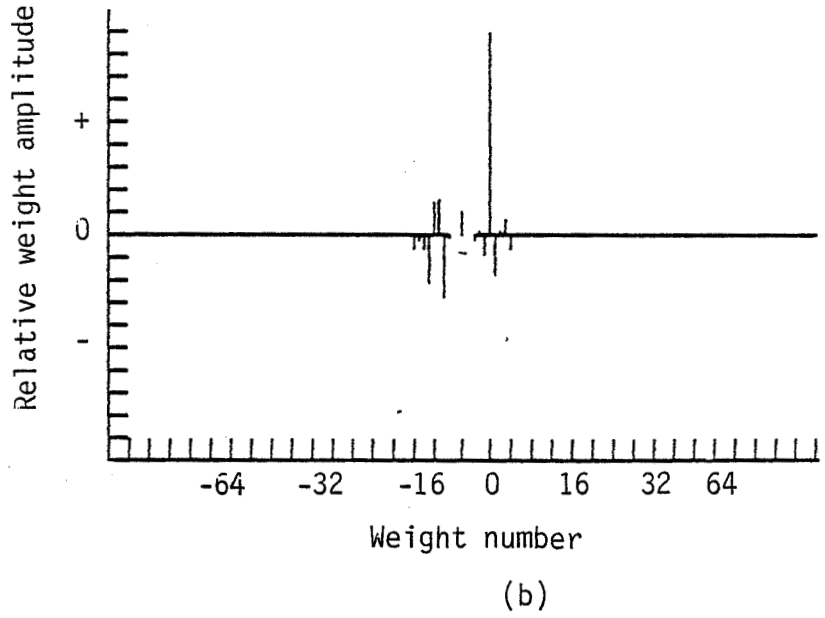


Figure 4.7. C=200. No improvement over C=10. (D=4, N=17, L=128, SNR=1). (a) TDE vs. number of iterations. (b) Final weight vector.

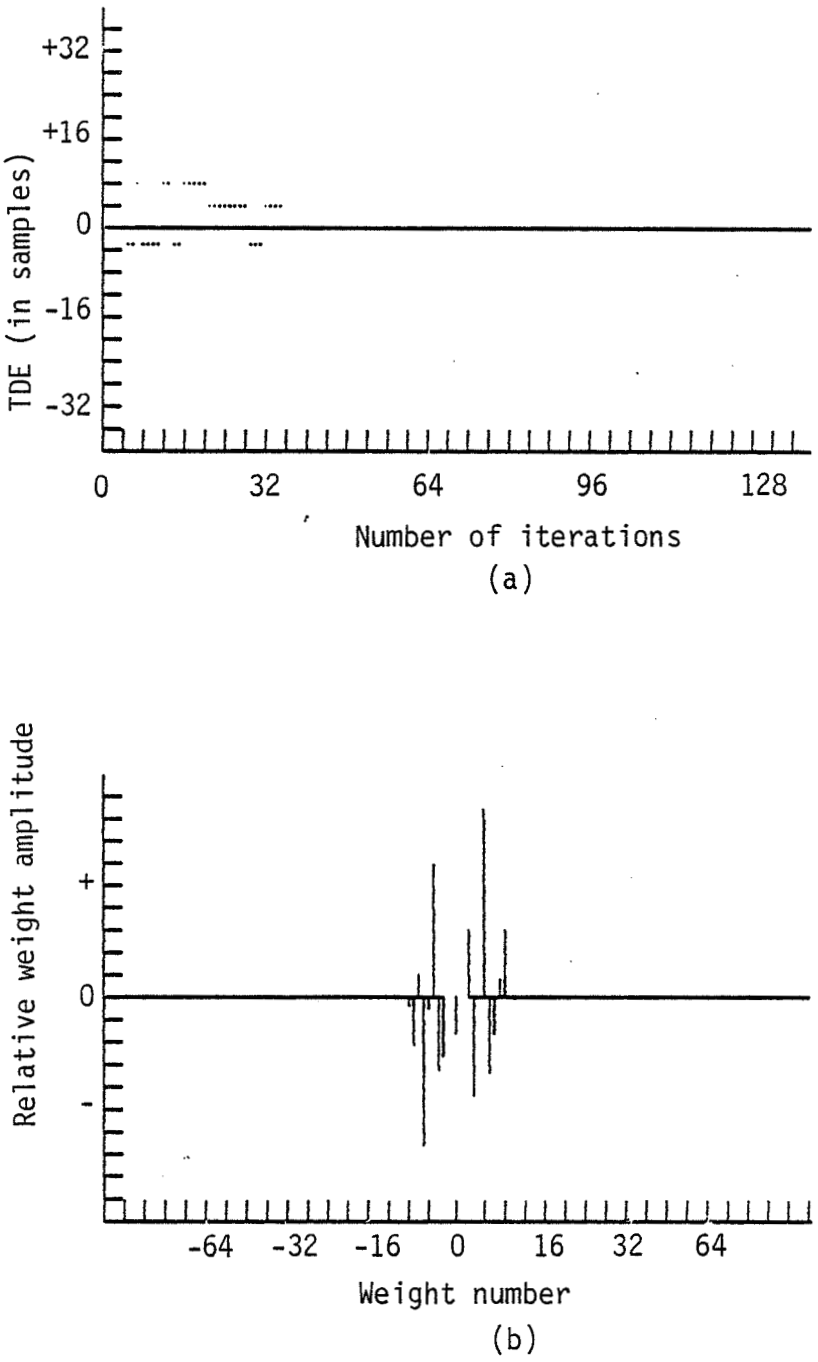


Figure 4.8. L=32. Ambiguity in filter output due to short record length. (D=4, N=17, SNR=1). (a) TDE vs. number of iterations. (b) Final weight vector.

response, after a satisfactory convergence, was obtained for L=128 in Figure 4.2. A larger number of iterations produced slightly improved results, though not appreciable, for the additional computation. This aspect is shown in Figure 4.9 for L=512 points.

4.2.3 Signal Parameters

This subsection illustrates the system response as the characteristics of the filter inputs are changed. The characteristics considered here are signal-to-noise power ratio (SNR), signal frequency bandwidth, and time varying delays.

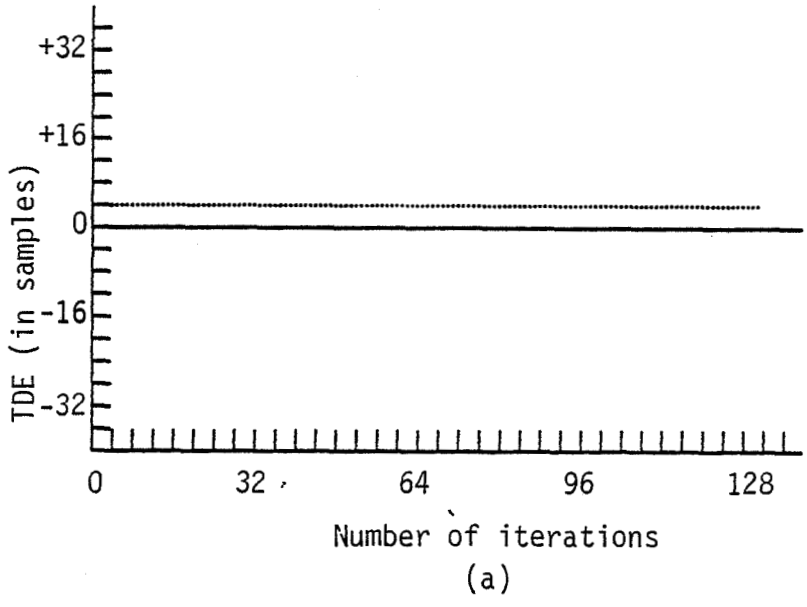
The signal to noise ratio used in the nominal response of Figure 4.2 was SNR=1. The response is evaluated in the next two figures for SNR =.5 and SNR=2. The input sequences were generated as

$$x(n) = SxR_1(n) + R_2(n)$$

$$d(n) = SxR_1(n+4) + R_3(n)$$

where $x(n)$ is the primary input to the filter and $d(n)$ is the secondary one. $R_1(n)$, $R_2(n)$, and $R_3(n)$ are uncorrelated random sequences with the same amplitude. The scale factor S equals .5 in Figure 4.10 and 2 in Figure 4.11. In all three cases, the response of the system was satisfactory after 128 iterations. However, as can be seen from the figures, the lower the SNR the slower the response (convergence).

The next simulation demonstrates the dependency of the main lobe



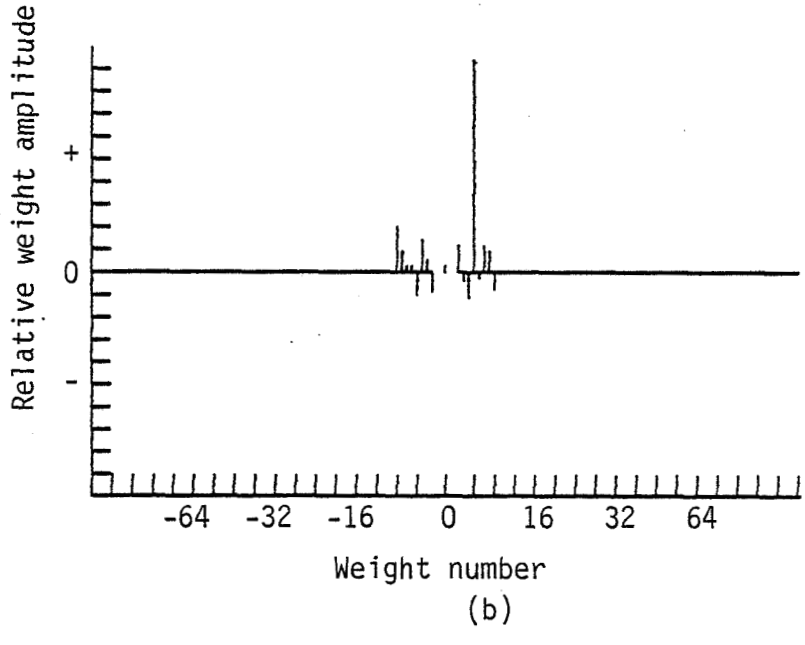
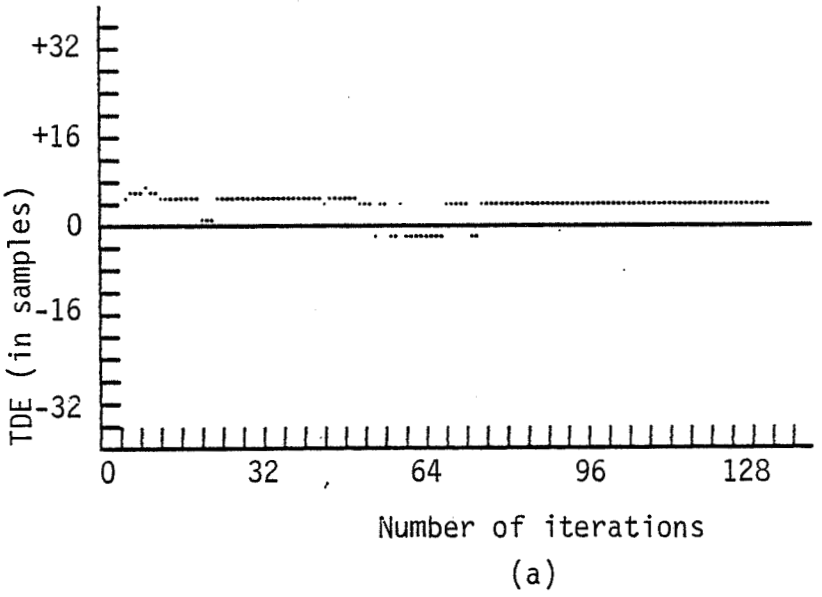
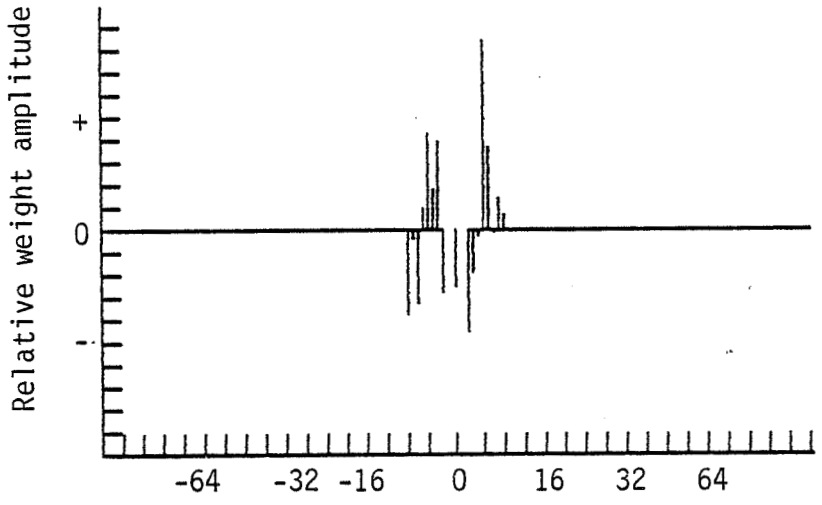


Figure 4.9. L=512. Improved response over L=128 but not necessary. (D=4, C=10, N=17, SNR=1). (a) TDE vs. number of iterations. (b) Final weight vector.

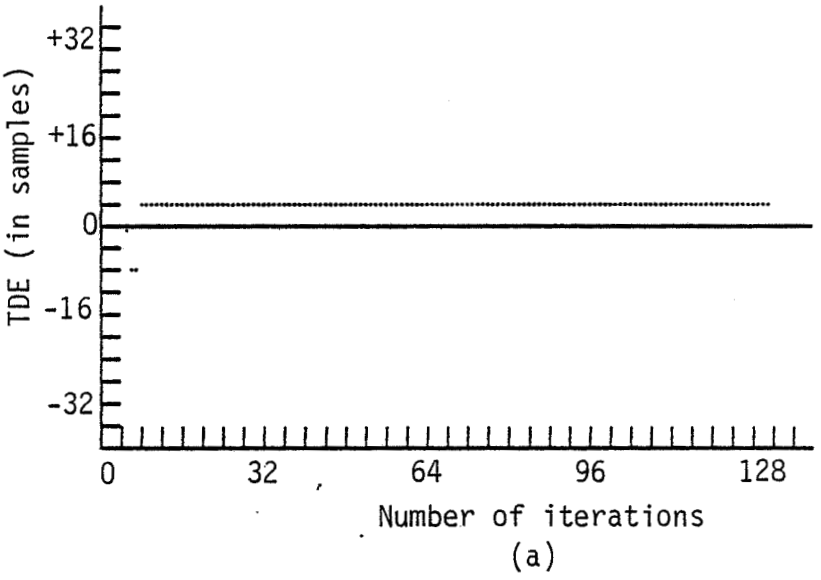


(a)



(b)

Figure 4.10. SNR=1/2. Slow Convergence. (D=4, C=10, N=17, L=128). (a) TDE vs. number of iterations. (b) Final weight vector.



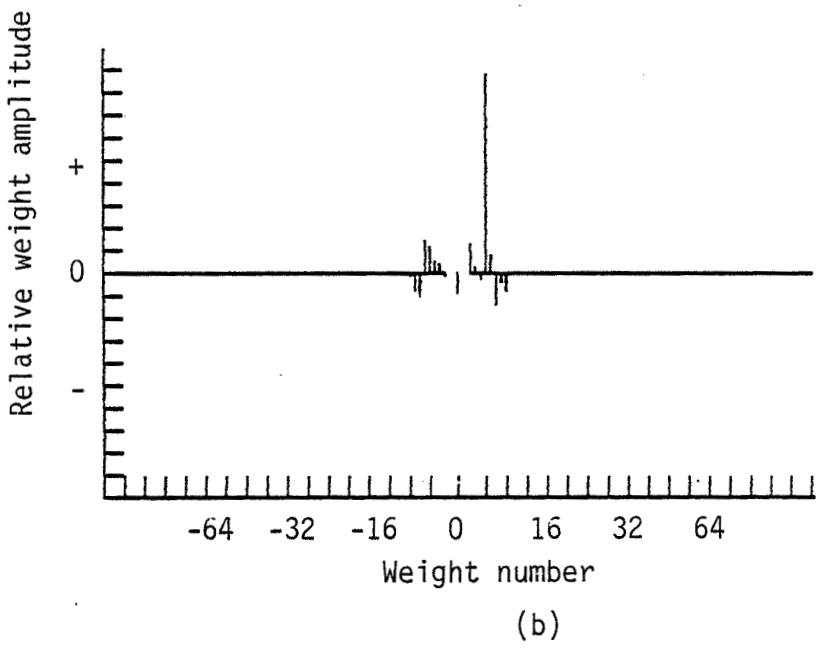


Figure 4.11. SNR=2. Faster convergence. (D=4, C=10, N=17, L=128).
(a) TDE vs. number of iterations. (b) Final weight vector.

width of the weight vector on the signal frequency band. The expression for the weight vector for an ideal band-limited signal and white additive noise was derived by Ahmed and Carter [17] as

$$W_{LMS}(K) = 4\beta\omega_b \, \mathrm{sinc}[\omega_b(K-D)] \cos[\omega_c(K-d)]$$

where β is a real constant, $\omega_b = \frac{\omega 1 - \omega 0}{2}$, $\omega_c = \frac{\omega 1 - \omega 0}{2}$. ω_1 and $\omega 0$ are the signal frequency limits, i.e. $G_{ss}(\omega) = 0$ for $\omega < \omega_0$ and $\omega > \omega_1$, where $\omega_1 > \omega_0$. It was shown in section 2.5 that the width of the main lobe of the weight vector is given by $\frac{\pi}{\omega_c}$, which becomes $\frac{2\pi}{\omega_1}$ for $\omega_0 = 0$. The input sequences were generated as

$$\chi(n) = S \times R_1(n) + R_2(n)$$

$$d(n) = S \times R_1(n+4) + R_3(n)$$

where $\chi(n)$, $d(n)$, $R_2(n)$, and $R_3(n)$ are the same as before, S is a scale factor used to obtain a SNR=1, and $R(n)$ is an ideally filtered sequence specified by its amplitude spectrum as

$$R_1(\omega) = \begin{cases} 1 \text{ for } 0 \leq \omega < \omega_1 \\ 0 \text{ for } \omega_1 \leq \omega \leq \pi \end{cases}$$

In Figure 4.2, the frequency band of $R_1(n)$ was given by $\omega_0 = 0$ and $\omega_1 = \pi$, i.e. white sequence. The main lobe in that case was 2

resolution $(\frac{2\pi}{\omega_1} = 2)$ wide. For the simulation in Figure 4.12, the bandwidth of $R_1(n)$ is limited to $\frac{\pi}{4}$, i.e.

$$G_{R_1 R_1}(\omega) = 0 \quad \text{for} \quad \frac{\pi}{4} < \omega \leq \pi \quad \rightarrow \quad \omega_0 = 0 \quad \text{and} \quad \omega_1 = \frac{\pi}{4}.$$

As seen in the figure the main lobe width now is $\frac{2\pi}{\frac{\pi}{4}} = 8$ resolutions.

Finally, $\omega_0 = 0$ and $\omega_1 = \frac{\pi}{8}$ were used in Figure 4.13. Producing a main lobe that is 16 resolutions wide.

It was mentioned in section 2.6 that one of the advantages of the LMSTDE algorithm over other methods is that it has the ability to track moving sources. The routine is tested here for this ability by linearly varying the delay between the input signals in a noisy environment. The input sequences were generated as

$$x(n) = R_1(n) + R_2(n)$$

$$d(n) = R_1(n+D(n)) + R_3(n)$$

where $R_1(n)$, $R_2(n)$ and $R_3(n)$ are uncorrelated random sequences as before, and $D(n)$ is a time varying delay. The signal-to-noise ratio used here is 2. In Figure 4.14, with $D(0) = 0$, $D(n)$ was increased by one every 64 points in 896-point input sequences. The solid line

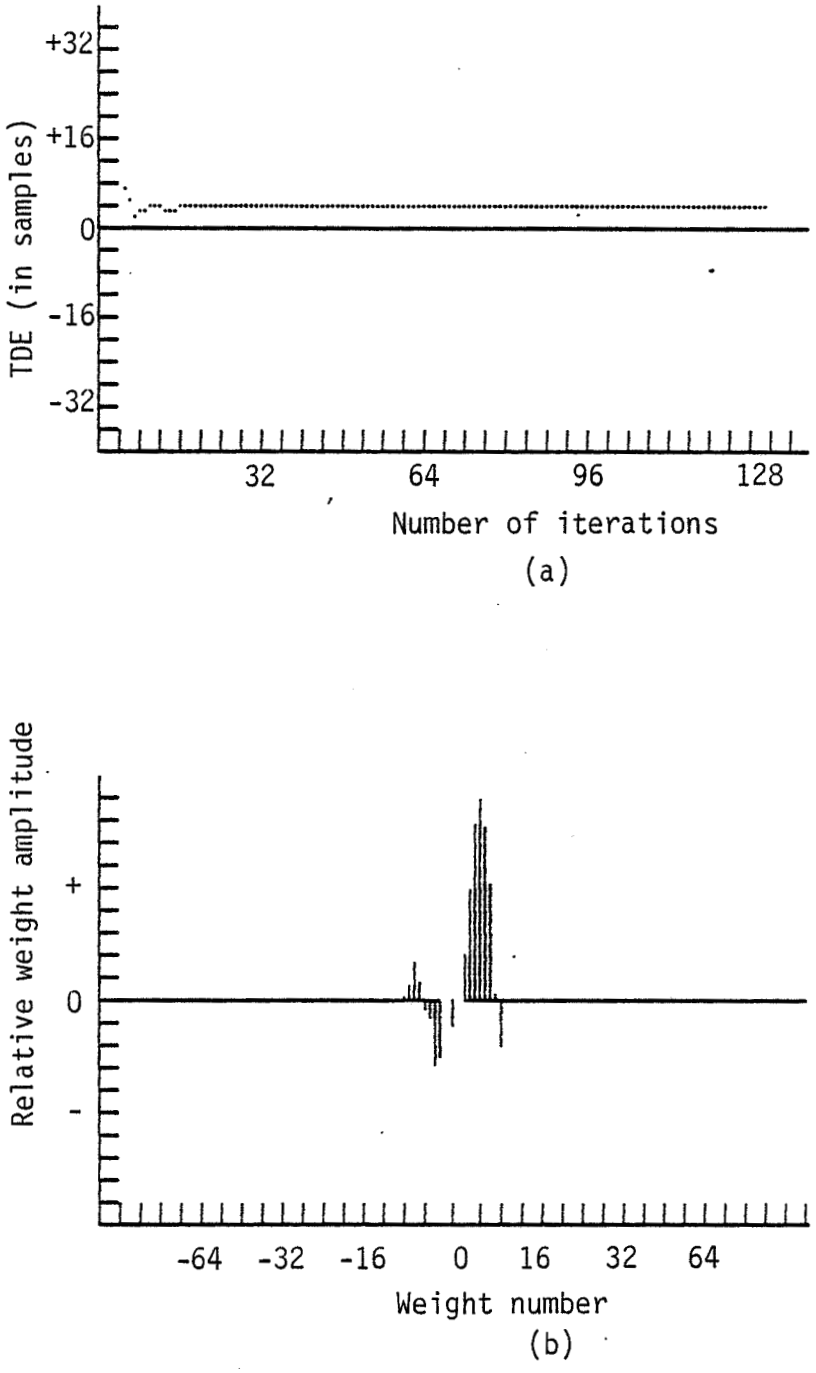


Figure 4.12. $W_1 = \frac{\pi}{4}$. Main lobe width of 8 resolutions. (D=4, C=10, N=17, L=128, SNR=1), (a) TDE vs. number of iterations. (b) Final weight vector.

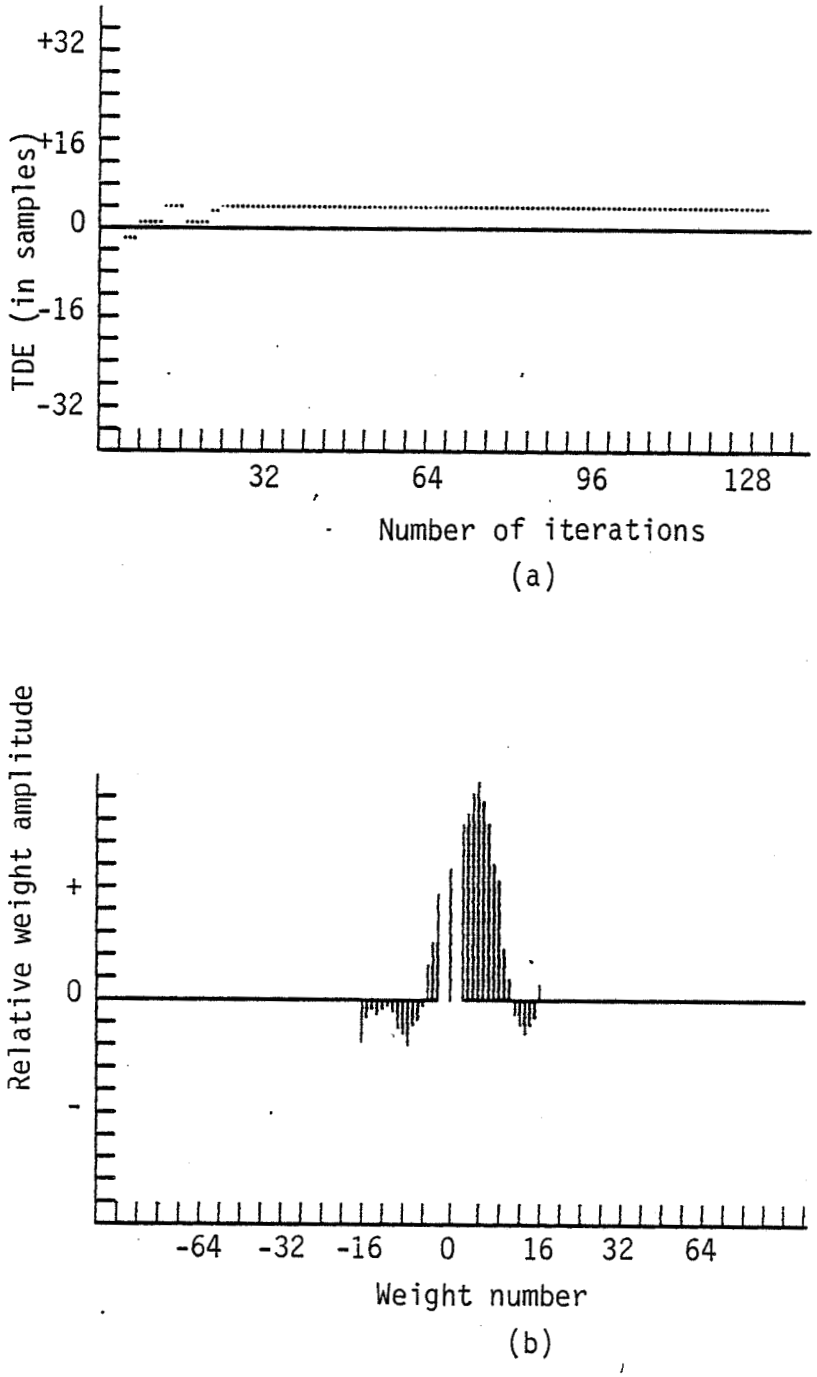


Figure 4.13. $W_1 = \frac{\pi}{8}$. Main lobe width of 16 resolutions. (D=14, C=10, N=33, L=128, SNR=1). (a) TDE vs. number of iterations. (b) Final weight vector.

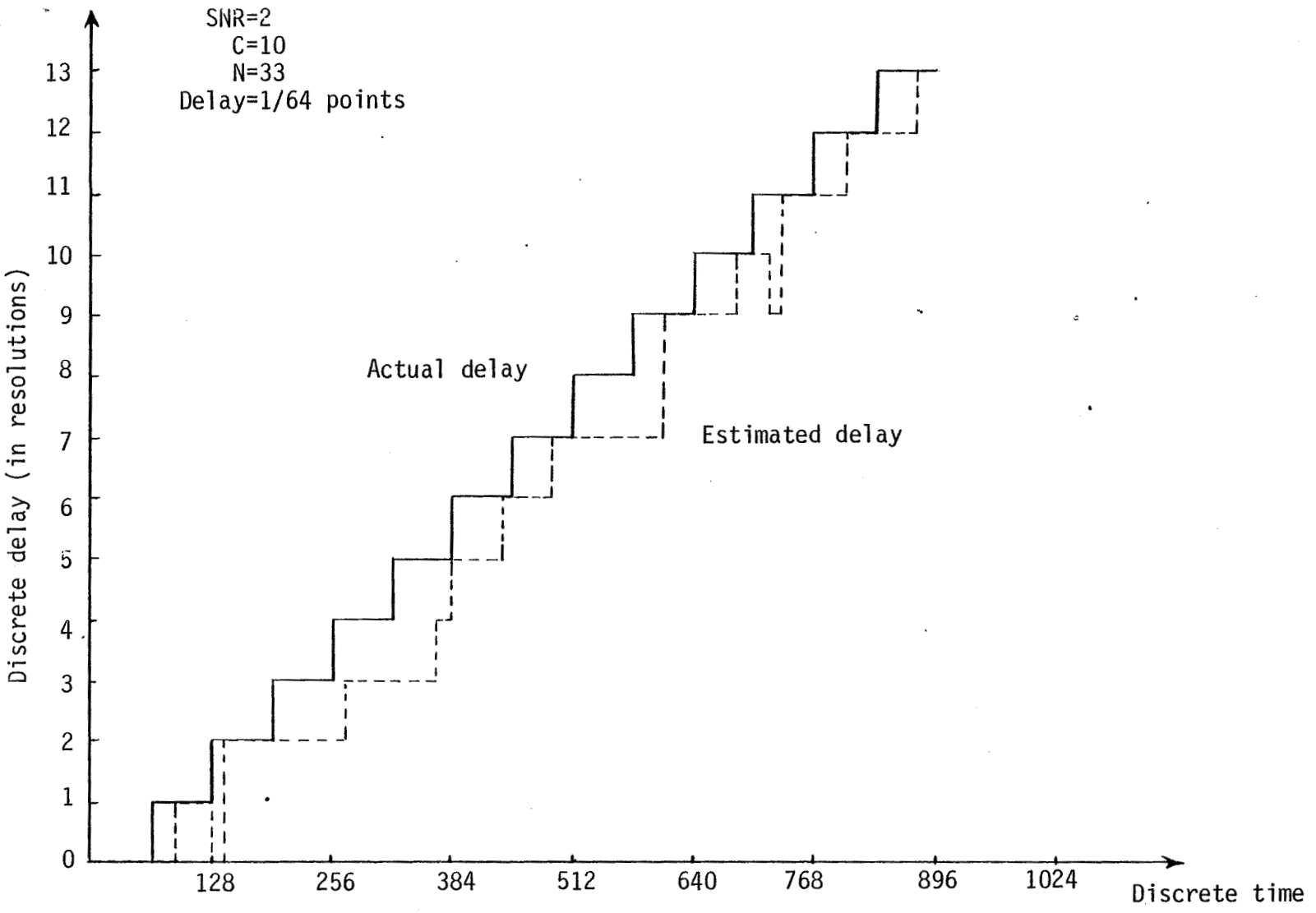


Figure 4.14 Filter output for time varying delay (1 point per 64 points).

represents the actual delay embedded, while the dashed line represents the instantaneous estimate of the delay as the abscissa value of the peak in the weight vector during 896 iterations. Figure 4.15 differs from Figure 4.14 in that D(n) is varied at the rate of 1 point every 128 points instead of 64. The adaptive filter tracked more closely the actual delay in the second case. This is to be expected, since the weights were given more time to converge. In both cases however, D(n) represents a relatively fast varying delay simulating a fast moving source. Considering this fact, one can conclude that the system response is acceptable and that the time lag, predicted by Feintuch, Bershad, and Reed [18], between the estimate and the actual delay is not as severe as it appears.

4.2.4 Shuttle Launch Noise

Long-range infrasound signals produced by a large rocket launch have been recorded and studied by several investigators [22]. These signals are similar in their statistical characteristics to weather-related infrasounds. Thus, they can be used to simulate the anticipated infrasounds generated by a far-field meteorological event. In this subsection, the adaptive filter algorithm is applied to the space shuttle launch signals and the overall response is evaluated.

Space Shuttle VIII was launched from Cape Canaveral, Florida, at 2:31 a.m. EST on 30 August 1983. About 52 minutes later the launch signal arrived at Wallops Island, Virginia, and was sampled on an array

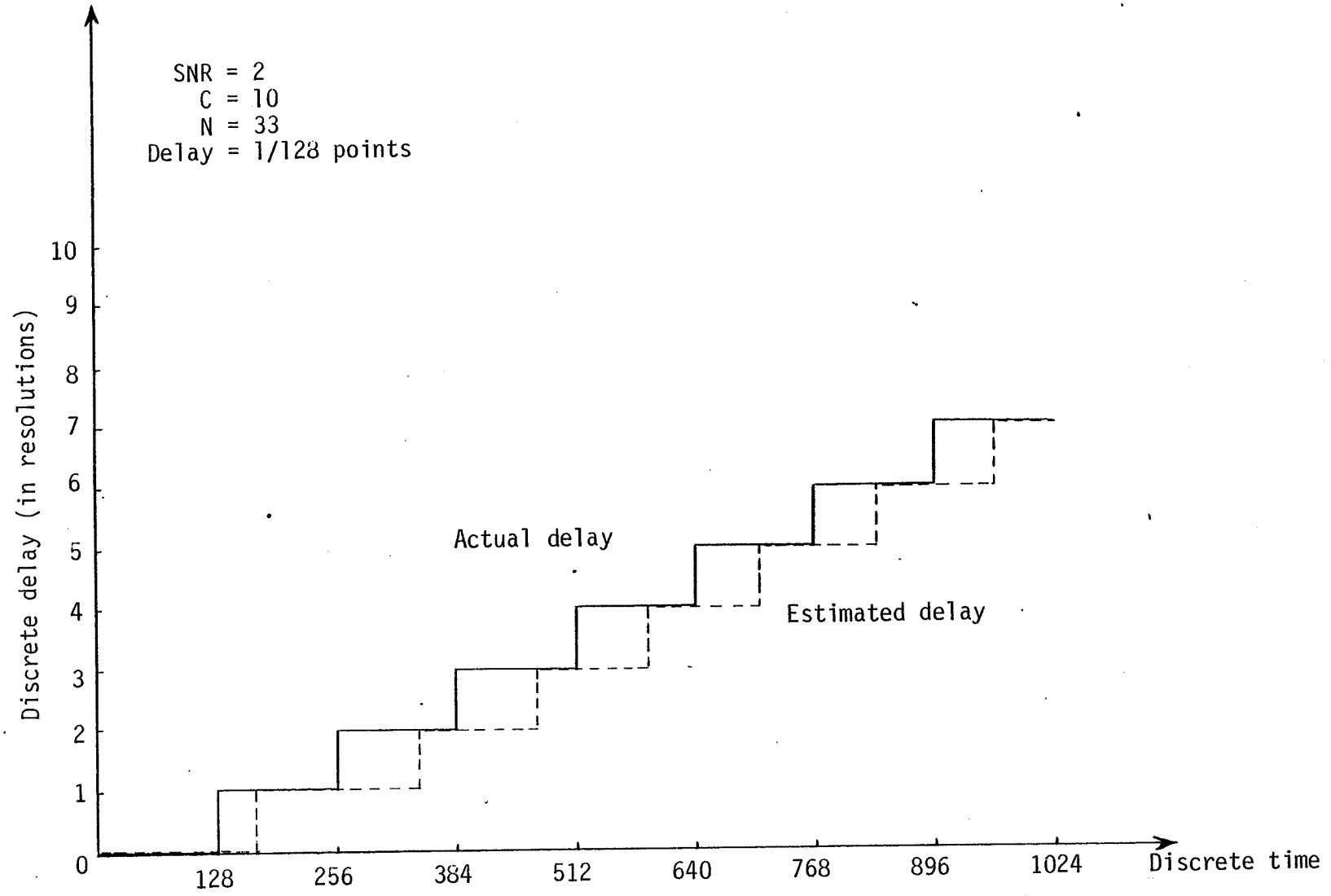


Figure 4.15. Filter output for time varying delay. (1 point per 128 points).

of four infrasonic microphones at the rate of 14 Hz. The array configuration is illustrated in Figure 4.16. Time histories and auto-power spectrums, of one channel before, during, and after the event, are presented in Figure 4.17 and Figure 4.18, respectively. These two figures represent a verification of the event and show the background of the signal that can be considered as the uncorrelated noise on the different microphones. The adaptive filter algorithm was applied to three pairs of channels during a 9.14 sec. (128 points) time frame. The output of the microphones during this time frame is plotted in Figure 4.19. The adaptive filter output for C2 x C3 (channel 2 cross channel 3), C3 x C4, and C4 x C2 is shown in Figure 4.20, Figure 4.21, and Figure 4.22, respectively. Making the estimate of the delays based on a 48-point, zero variance, window in the TDE vs. time plots yields the following results:

C2 leads C3 by .455 sec. (+7 resolutions)

C3 lags C4 by .780 sec. (-12 resolutions)

C4 leads C2 by .325 sec. (+5 resolutions)

Based on these delays and on the array configuration, a source direction of about 210° from north can easily be obtained using trigonometric identities. This estimate of the bearing is to be compared with the true direction of 207° as illustrated in Figure 4.23. The error in the

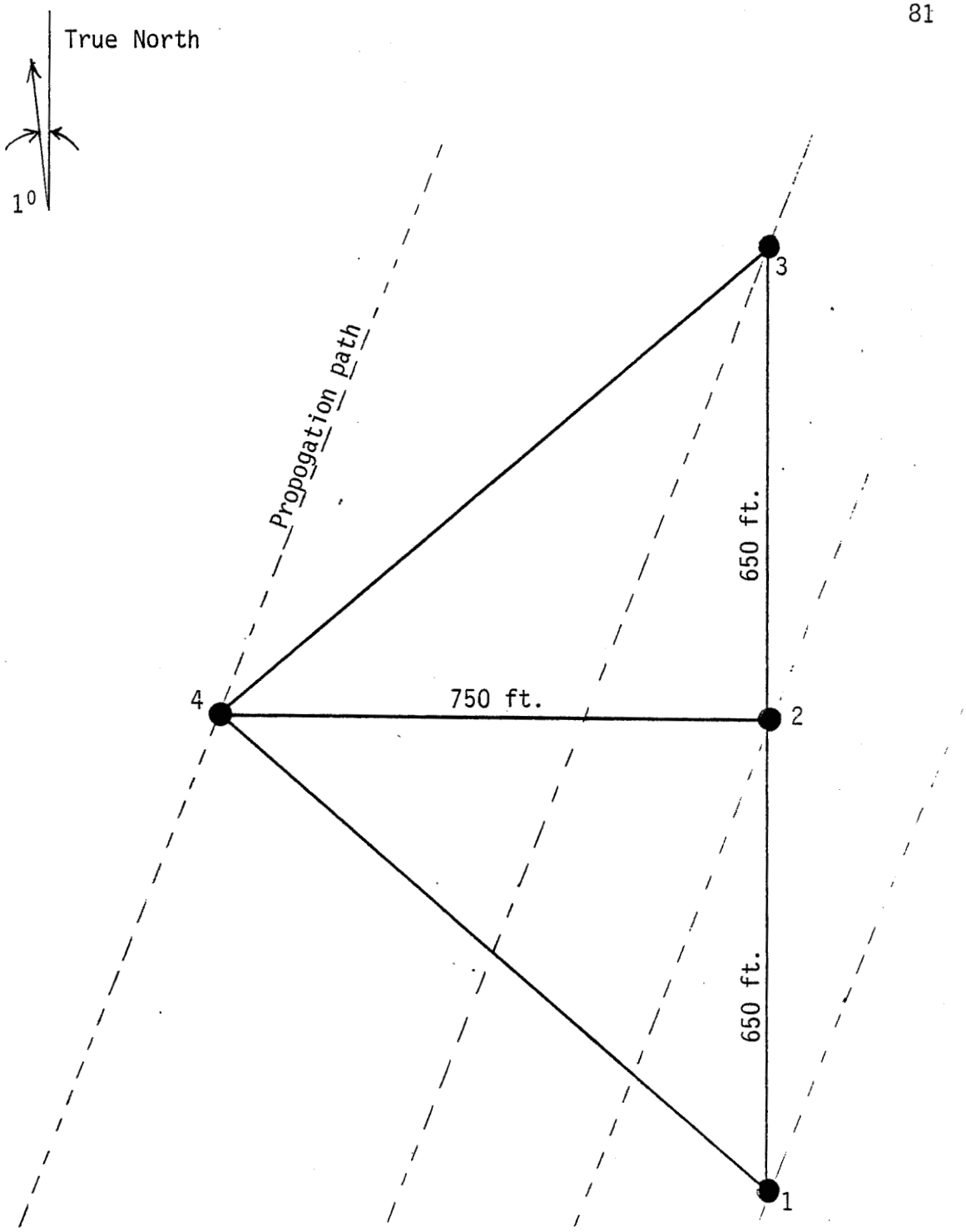


Figure 4.16. Array configuration used in sampling STS-8 launch signal.

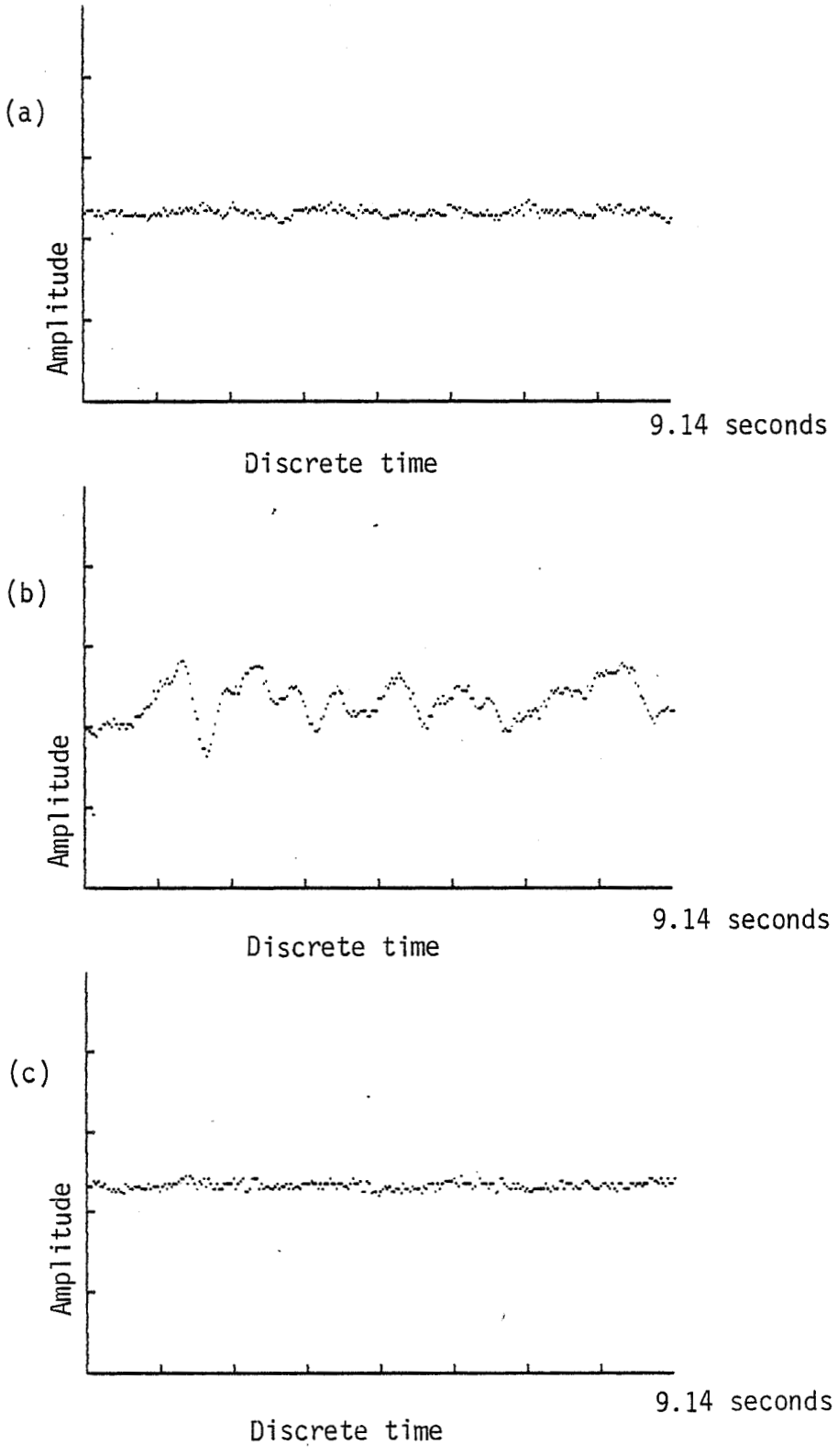


Figure 4.17. Time histories of STS-8 launch signal (in equivalent arbitrary units) on one channel, (a) before, (b) during, and (c) after the arrival of launch signal.

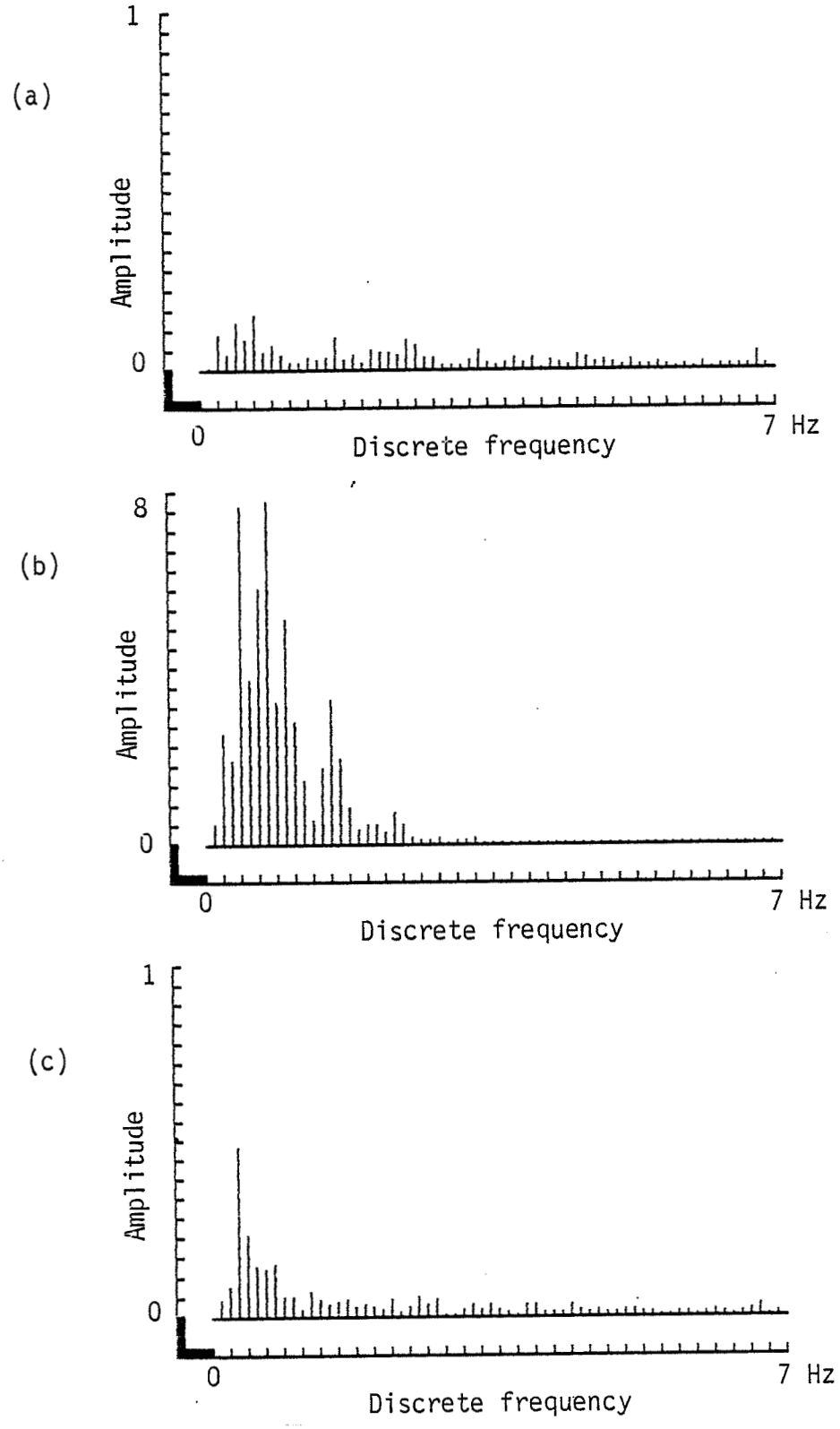


Figure 4.18. Auto-power spectrums of STS-8 launch signal (in arbitrary units) on one channel, (a) before, (b) during, and (c) after the arrival of launch signal.

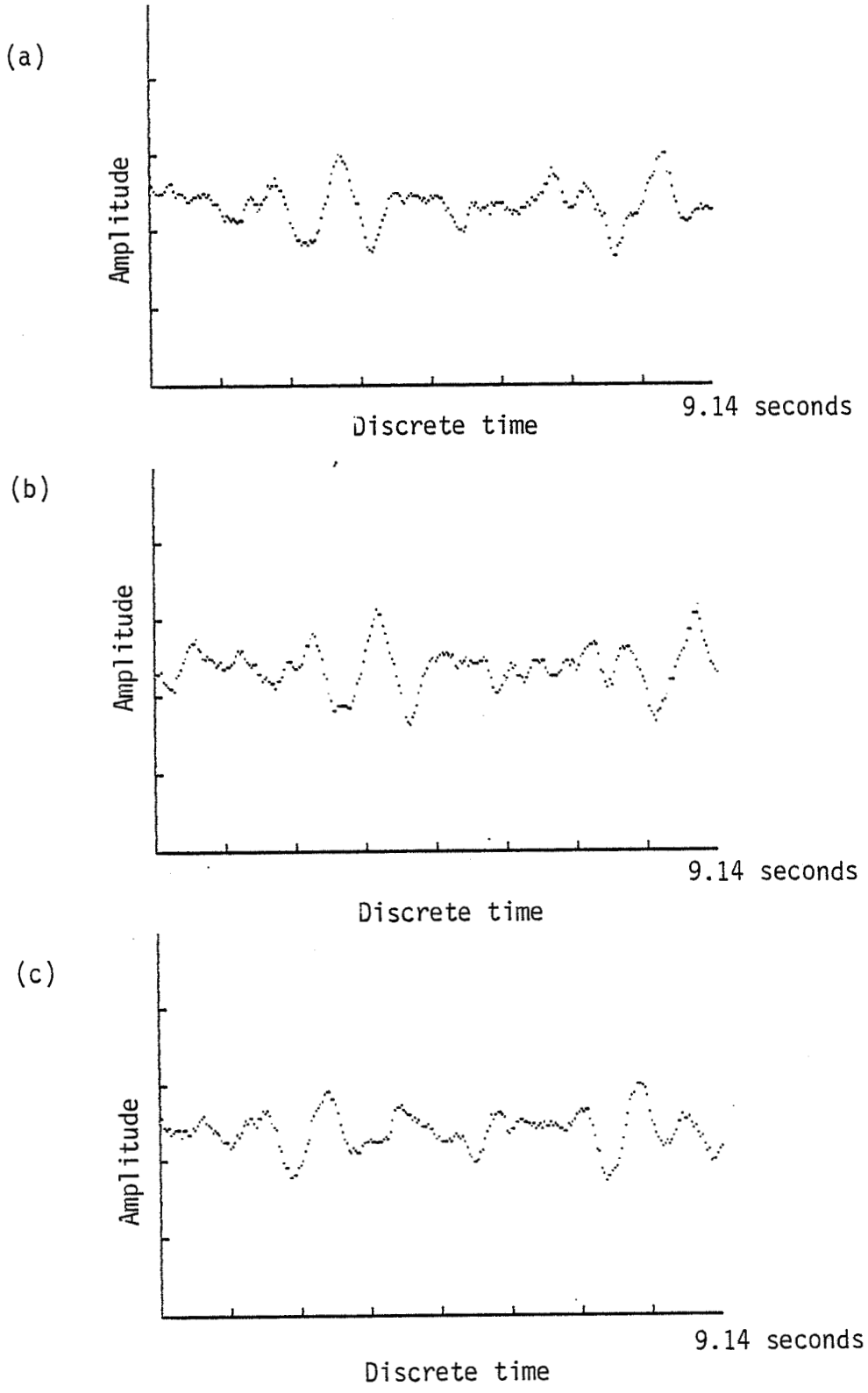


Figure 4.19. Time histories of STS-8 launch signal (in equivalent arbitrary units) on three microphones, (a) channel 2, (b) channel 3, (c) channel 4.

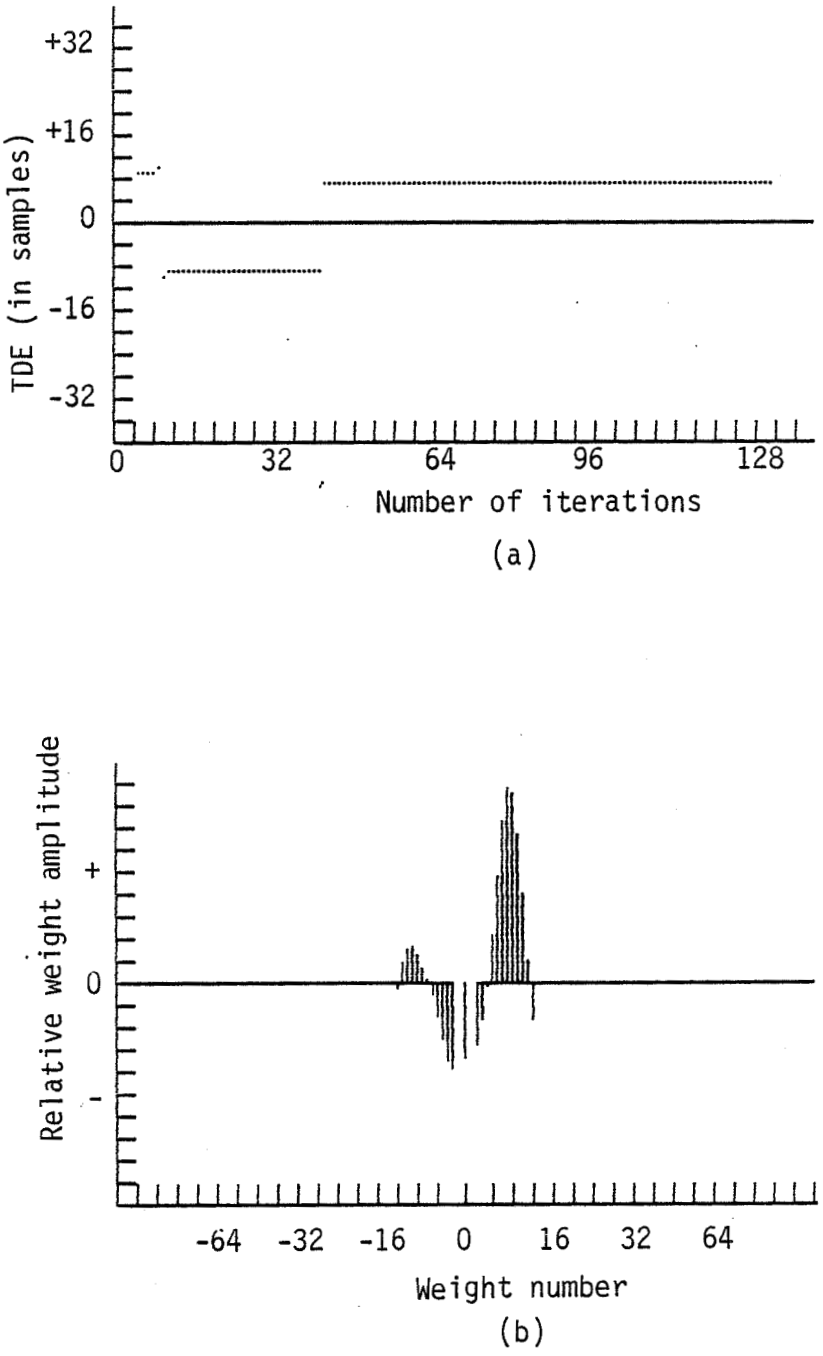


Figure 4.20. Filter response for launch signal, C2 x C3, (a) TDE vs. time, (b) Final weight vector.

C-2

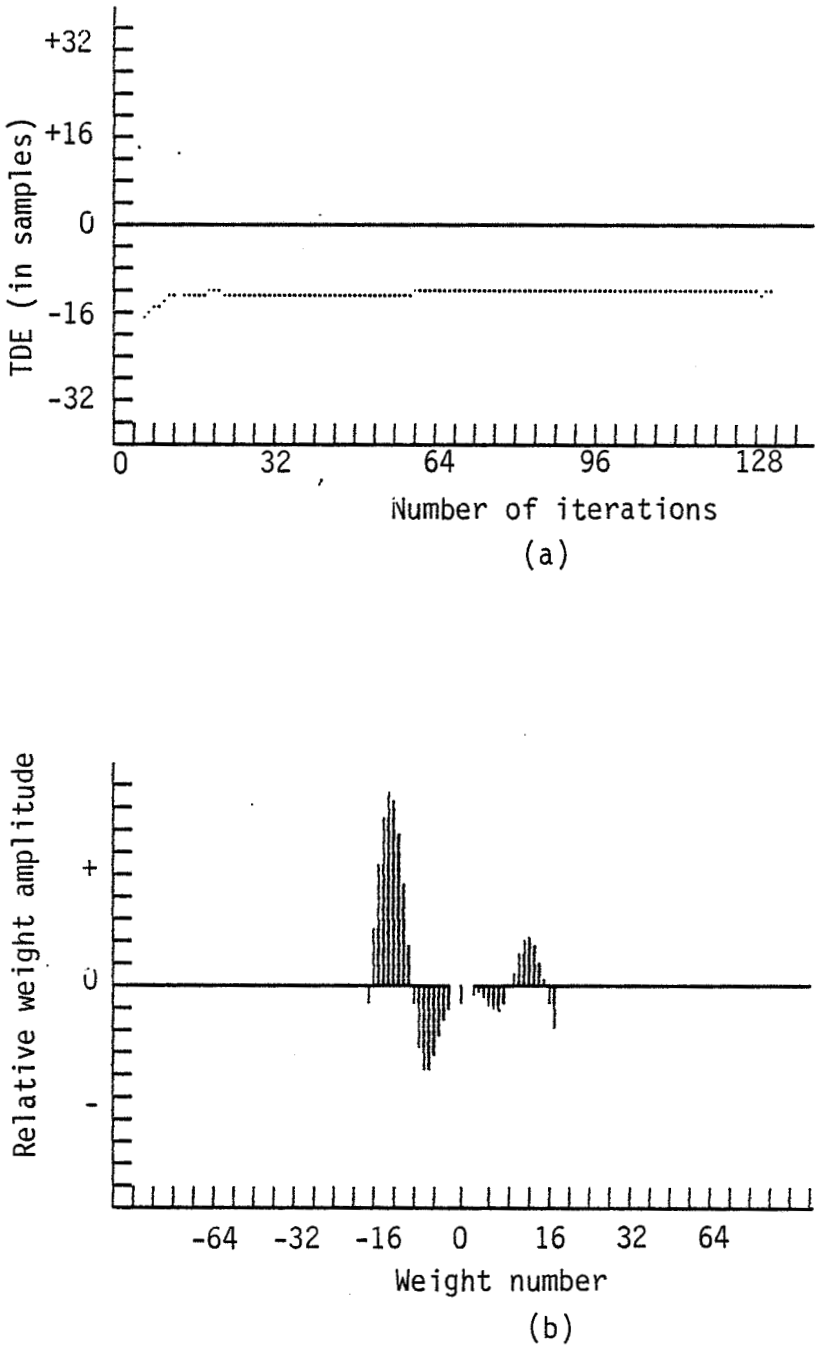


Figure 4.21. Filter response for launch signal, C3 x C4, (a) TDE vs. time, (b) Final weight vector.

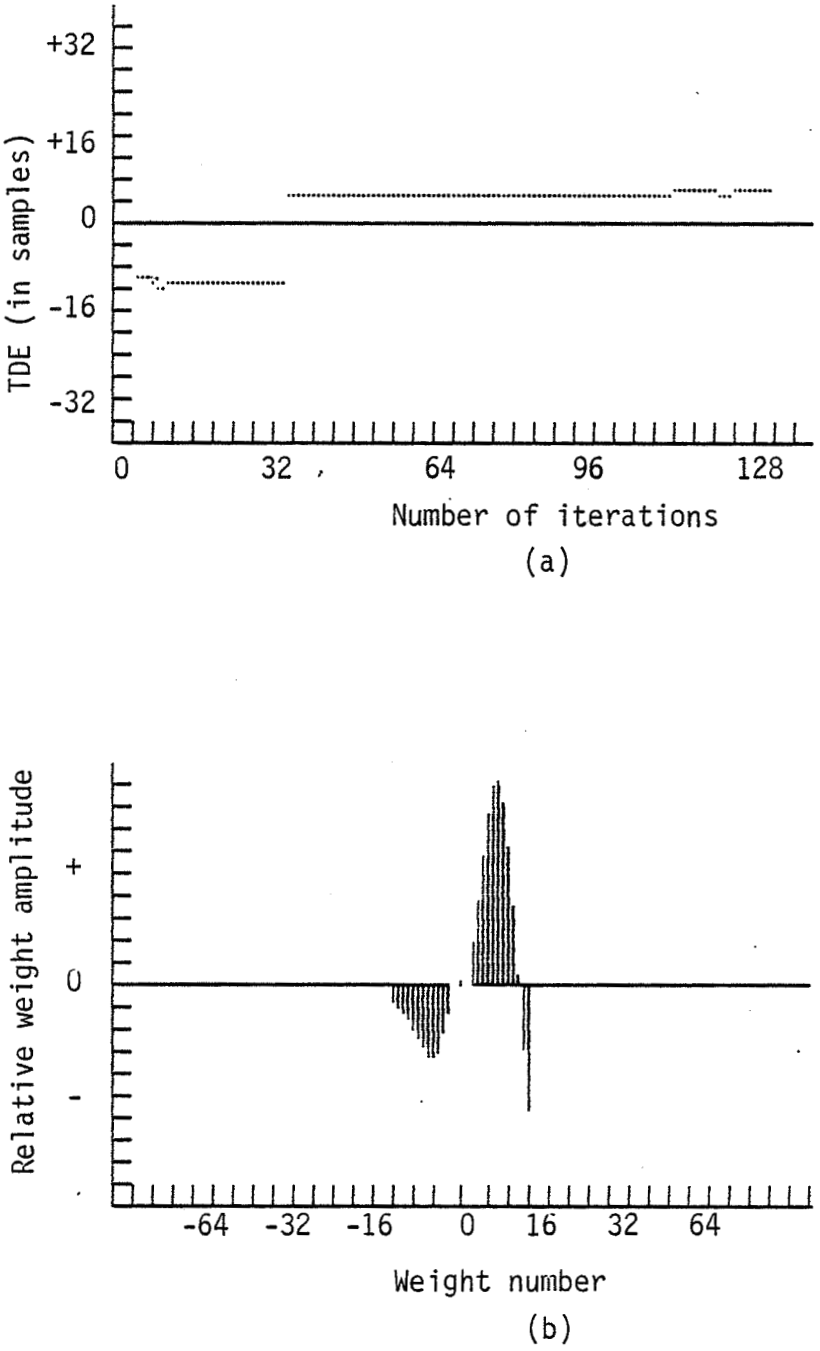


Figure 4.22. Filter response for launch signal, C4 x C2, (a) TDE vs. time, (b) Final weight vector.

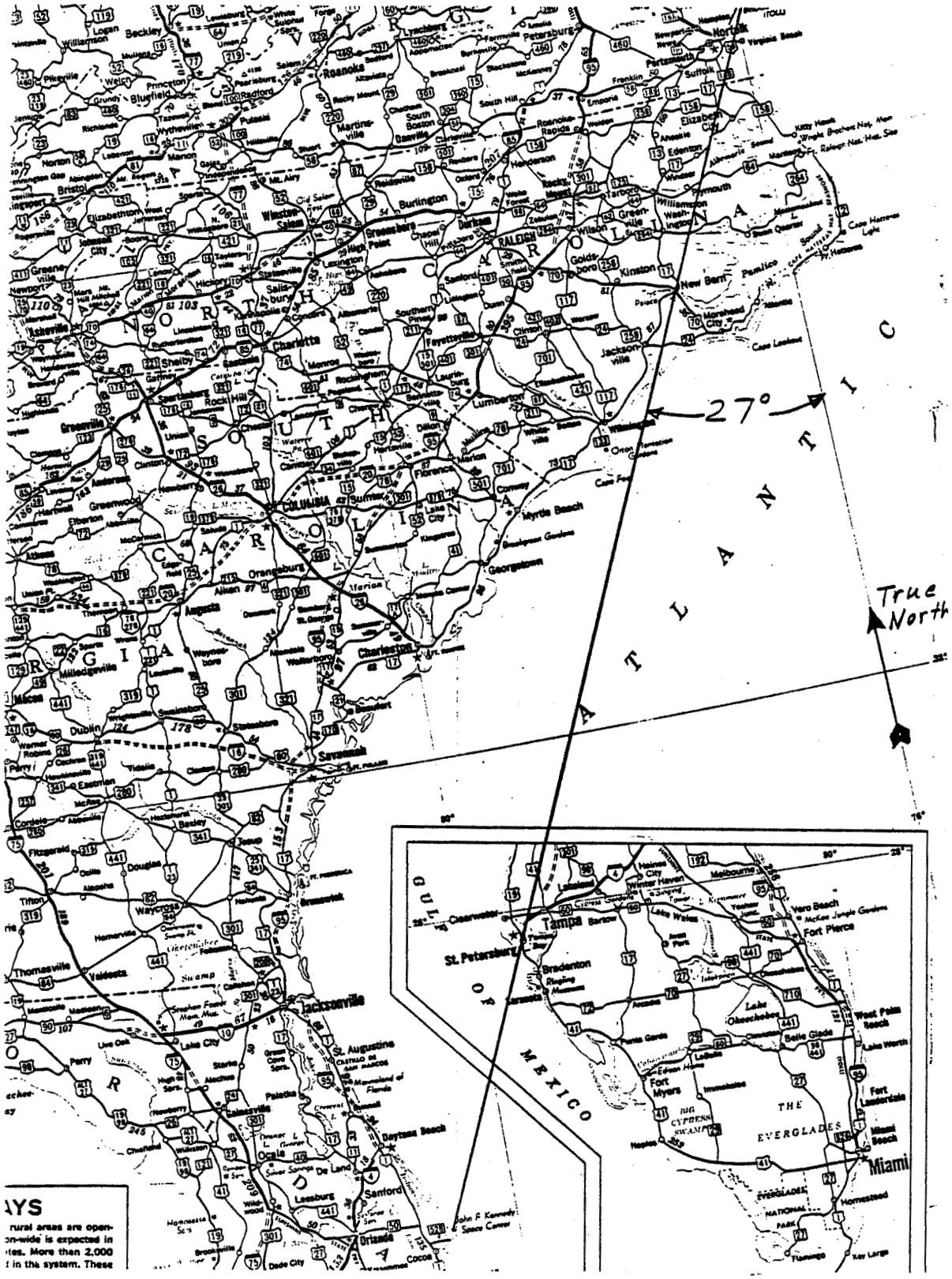


Figure 4.23. True direction of shuttle launch site.

estimate is within the bearing resolution. In this case, the bearing resolution (for f_s = 14 Hz and array foot of 650 ft) is about 6.8°.

4.3 System Operation and Results

The previous section demonstrated the effectiveness of the LMS adaptive filter algorithm in estimating the delay between the outputs of two sensors. The algorithm, as discussed in section 3.3, can now be used as a subroutine in a larger software package for a complete, actual field operation of the system described in section 3.4. This section presents a description of this software package, the mode in which it is operated, and the way the results are evaluated.

The software package is depicted in Figure 4.24. It consists of an automatic detection routine, as well as a bearing computation, of infrasonic sound waves. The whole process is automatic in the sense that no user intervention is needed after the program is started and the parameters are set. However, this automation is limited by the data storage capacity of the computer. The process is started by simultaneously sampling three 1024-point files from the three sensors* in the array. The data files are stored in temporary buffers until they are analyzed by the LMSTDE algorithm. The algorithm, as described in section 3.1, is then applied to each pair of data files at a time, considering only the first 128 points as the filter data record. If the convergence test, using the variance criterion, fails for any pair of sensor outputs, the process will be stopped and a new set of data files is sampled. When

*Only three out of the four sensors in the array are at this stage.

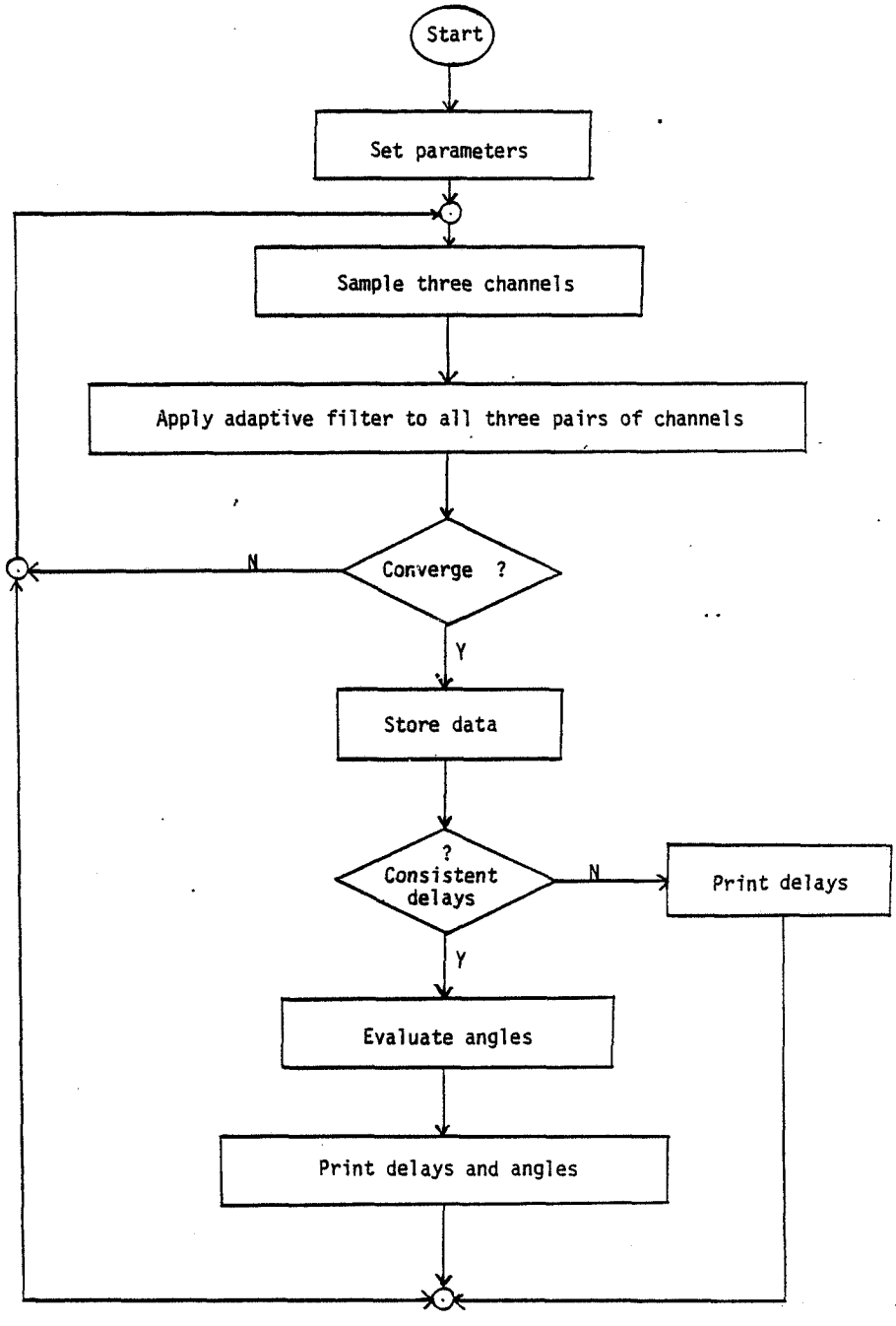


Figure 4.24. Software flowchart.

all three pairs of files show good coherent data through acceptable variance levels, the data files are transferred to a floppy disk for permanent storage.

As the detection phase ends, the spectral content of the signal, the bearing, and the nature of the source are then evaluated. The analysis begins by testing the delays, obtained from the LMSTDE algorithm, for consistency assuming plane wave propagation and 0° elevation. If the consistency test is not passed the delays are printed for later analysis; otherwise three angles, corresponding to the three delays, are evaluated using simple trigonometric relations. The delays and angles are then printed for the record, and the whole process is started over again.

The consistency test discriminates between plane waves and cylindrical, or spherical propagating waves. The later category may occur if the source is in the near field of the array (within one mile from the center of the array). Since the intention of this research is the early detection of far-field events, this category is neglected by the software. The delays obtained in such cases are only printed for direct analysis by the user. The trigonometric relations used to evaluate the bearing in case of a plane wave were formulated for horizontally incoming waves only. In this case, the three computed angles will be the same while they will differ from each other for a plane wave propagating with a non-zero elevation. The difference among the angles can be easily related to the value of the incident angle. The software does not produce this calculation. For further discussion see Appendix A.

The software has been applied since early June 1984 at NASA Langley Research Center. An average of ten data files have been stored daily, about half of which indicate plane wave propagation with printed delays and angles. This printed data is then compared with the weather reports, obtained from the meteorology center at NASA, on a daily basis. Finally, the data files stored on floppy disks will undergo a comprehensive spectral analysis, mainly for signature identification and power law verification. The software used for the spectral analysis is, or has stemmed from, DAISE (Digital Analysis of Infra-Sound Experiments) [23], that was developed previous to the course of this research.

Daily evaluation of the data proved that most files stored by the system can be grouped into four categories, with their typical sources, as follows:

	Man-made	Weather-related
Near-field*	Heavy machinery	Microbursts
Far-field	Military aircrafts	Severe storms

*Within one mile from the center of the array.

Man-made infrasounds were to be expected since the microphone array is located in Langley Research Center where a wide range of activities take place on a regular basis. These include operating all sorts of machinery, wind tunnels, etc. Future plans for near-field (both categories) signal analysis exist but they are not of interest at this point. Infrasounds produced by far-field events are discussed in the next two

sections. The man-made category is discussed in section 4.4 while the weather-related category is discussed in section 4.5.

4.4 Man-Made Signals

This section discusses the man-made signals with long-range propagation detected by the automatic detection routine described in the previous section. This category of infrasounds is of interest to this study for two reasons. First, it verifies the capability of the system to detect infrasonic signals and to locate their sources. Second, it illustrates the necessity of more sophisticated signal processing techniques to distinguish between these events and the meteorological ones.

Among others, the non-weather related signals detected include single-component infrasounds (for example 7.2 Hz, 10 Hz, 16.8 Hz), sonic booms generated by supersonic vehicles, and infrasounds from jet powered aircraft. Single component infrasounds have been detected daily. The system estimate of their directions places their sources in the main research center of Langley. Thus, it is believed that they are mostly related to the different activities in the Center. Few sonic booms were detected during the summer of 1984. They were mainly associated with flights of supersonic military aircraft such as the Air Force F-15 which are stationed nearby at Langley Air Force Base. The high altitude of these sonic booms was indicated by elevation angles of 25° to 40° evaluated from the data. The third type of man-made signal was found in several files over a period of about three months. The common characteristics between these infrasounds include a sound pressure level of

about 90 dB, frequency band of about 1 to 6.5 Hz, and a source direction ranging from 130° to 150°, relative to true north. The consistency of these values between files, the relatively strong amplitude, and the directions found all have led to the belief that these infrasounds are generated during the take-off of military jet powered aircraft from Langley Air Force Base. The location of the runway at the base relative to the microphone array is shown in Figure 4.25.

One of the jet data files is considered here for illustration. The file was stored on August 28, 1984, by the automatic detection routine. The parameters of the software and the hardware were set to their nominal values as specified in Table 3.1. The signal on one channel before and during the event is shown in Figure 4.26 in the time domain and in Figure 4.27 in terms of its auto-power spectrum. The signal on microphone 1, 2, and 3 during a different time frame, of the same file, is shown in Figure 4.28. To demonstrate the convergence of the algorithm for this type of data, the filter response was recomputed for C1 x C2 in Figure 4.29, C2 x C3 in Figure 4.30, and C3 x C1 in Figure 4.31. This response was recomputed by feeding alternative data points of the time histories to the adaptive filter. This effectively reduced the sampling rate to 25.6 Hz (51.2 Hz originally). A 25.6 Hz sample rate is adequate in this case since the signal has a narrow frequency band as can be seen in Figure 4.27. Keeping this rate in mind, examining the filter response yields the following results

C1 lags C2 by 0.742 sec. (-19 resolutions)

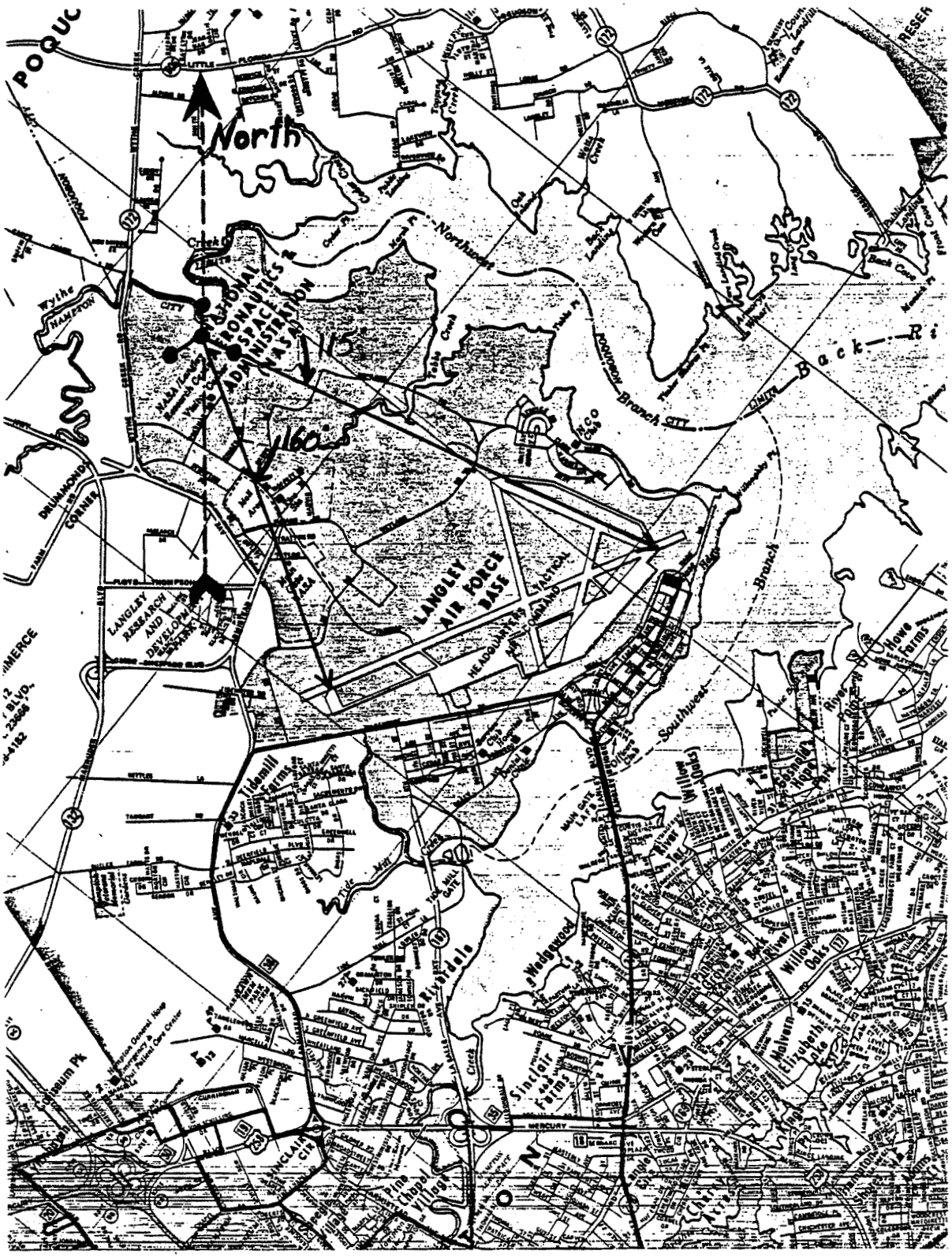


Figure 4.25. Location of the runway in Langley Air Force Base relative to the microphone array.

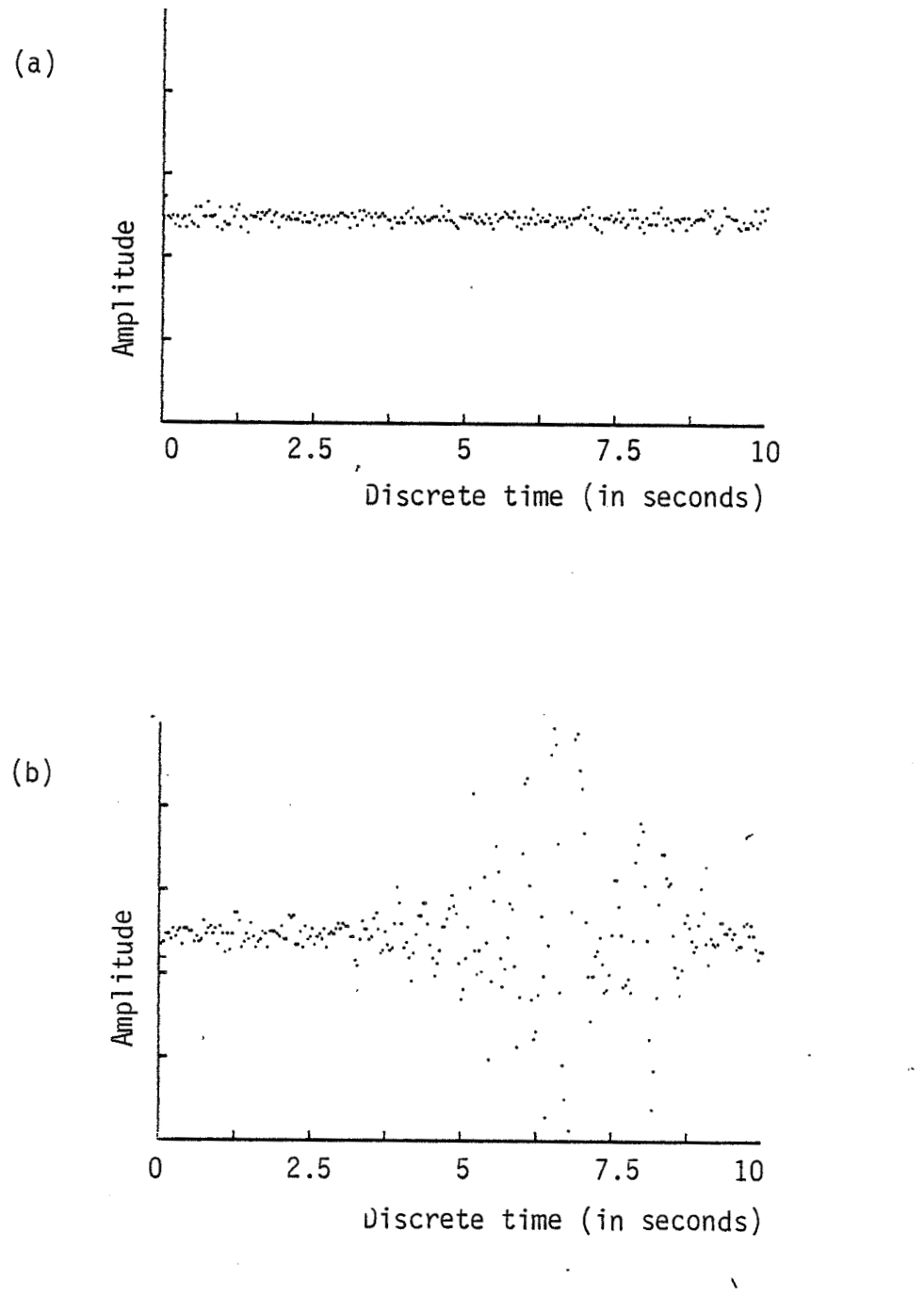


Figure 4.26. Time histories (in equivalent arbitrary units) of infrasounds generated by jet-powered aircraft, (a) before, and (b) during the event.

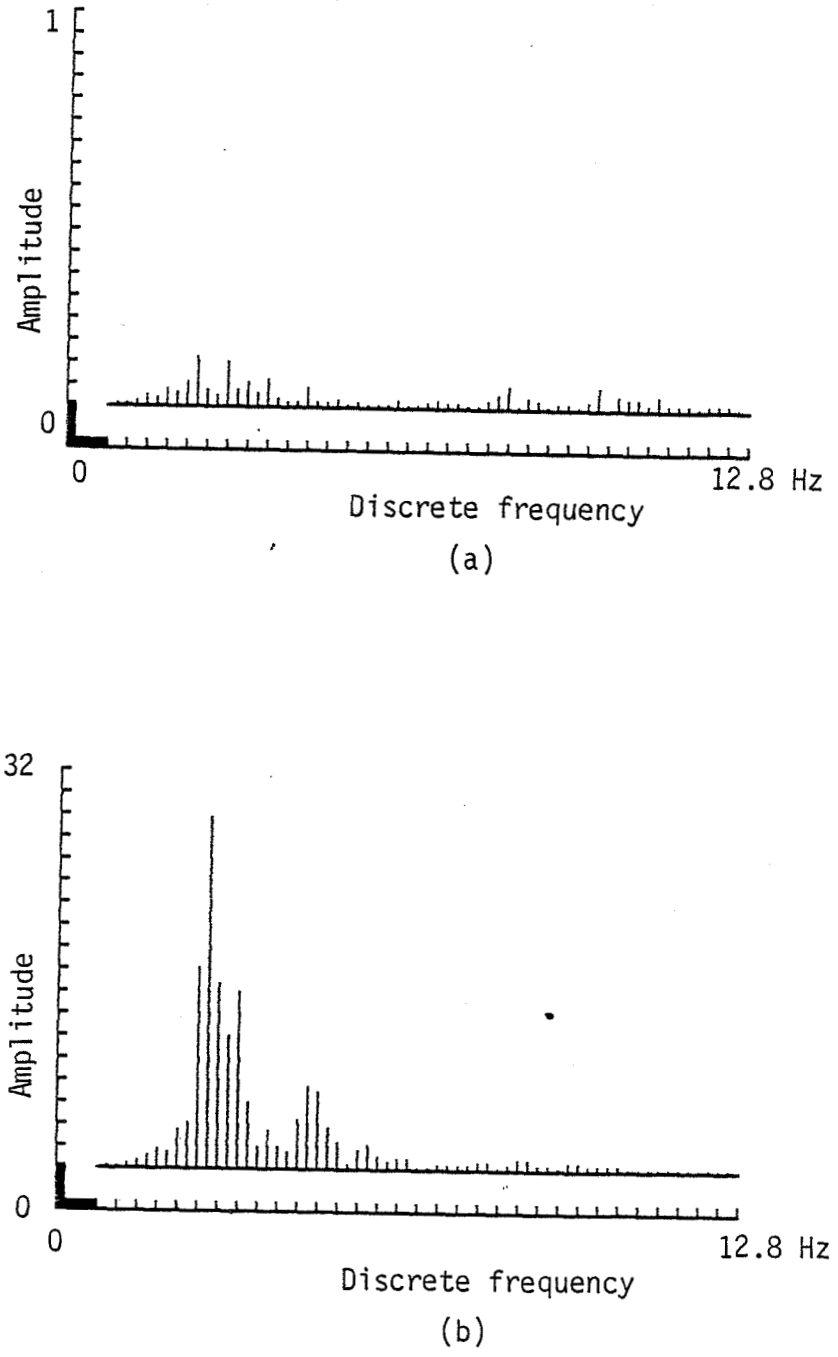


Figure 4.27. Auto-power spectrums (in arbitrary units) of infrasounds generated by jet-powered aircraft, (a) before, and (b) during the event.

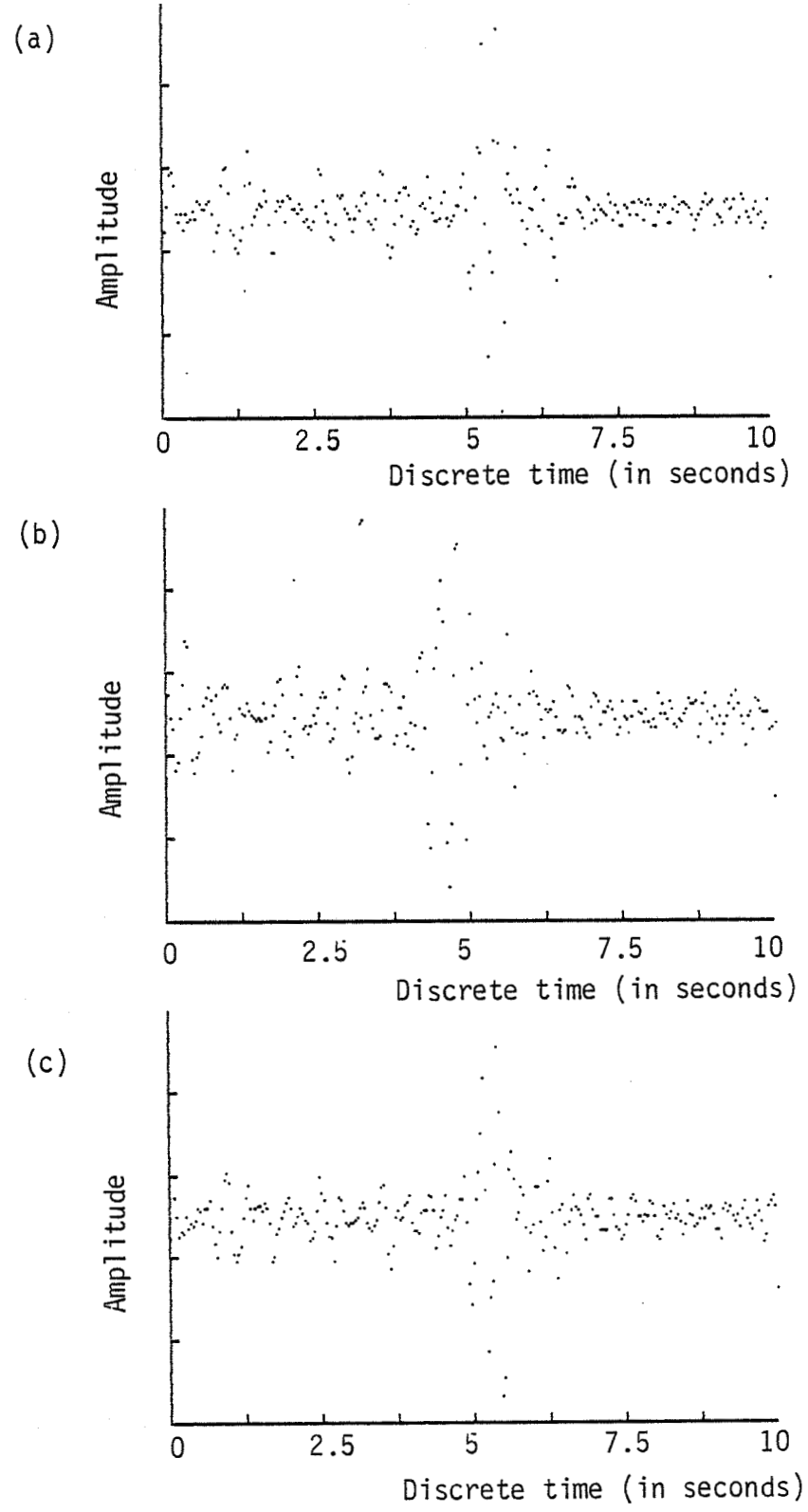
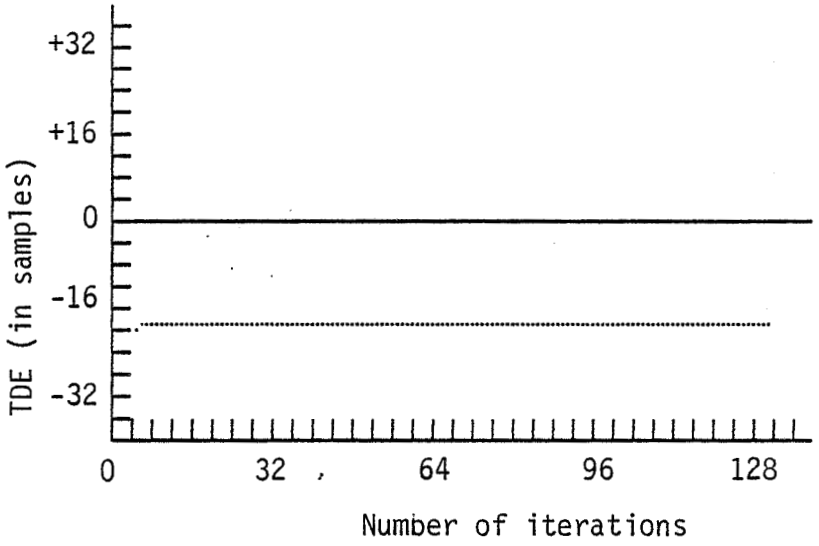
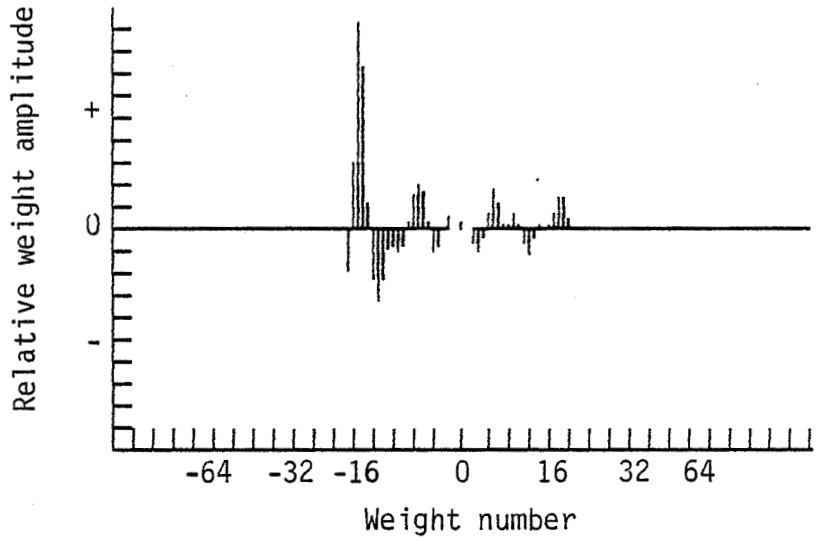


Figure 4.28. Time histories (in equivalent arbitrary units) of infrasounds generated by jet-powered aircraft, sampled on (a) channel 1, (b) channel 2, and (c) channel 3.



(a)



(b)

Figure 4.29. Filter response for jet data, C1 x C2, (a) TDE vs. time, (b) Final weight vector.

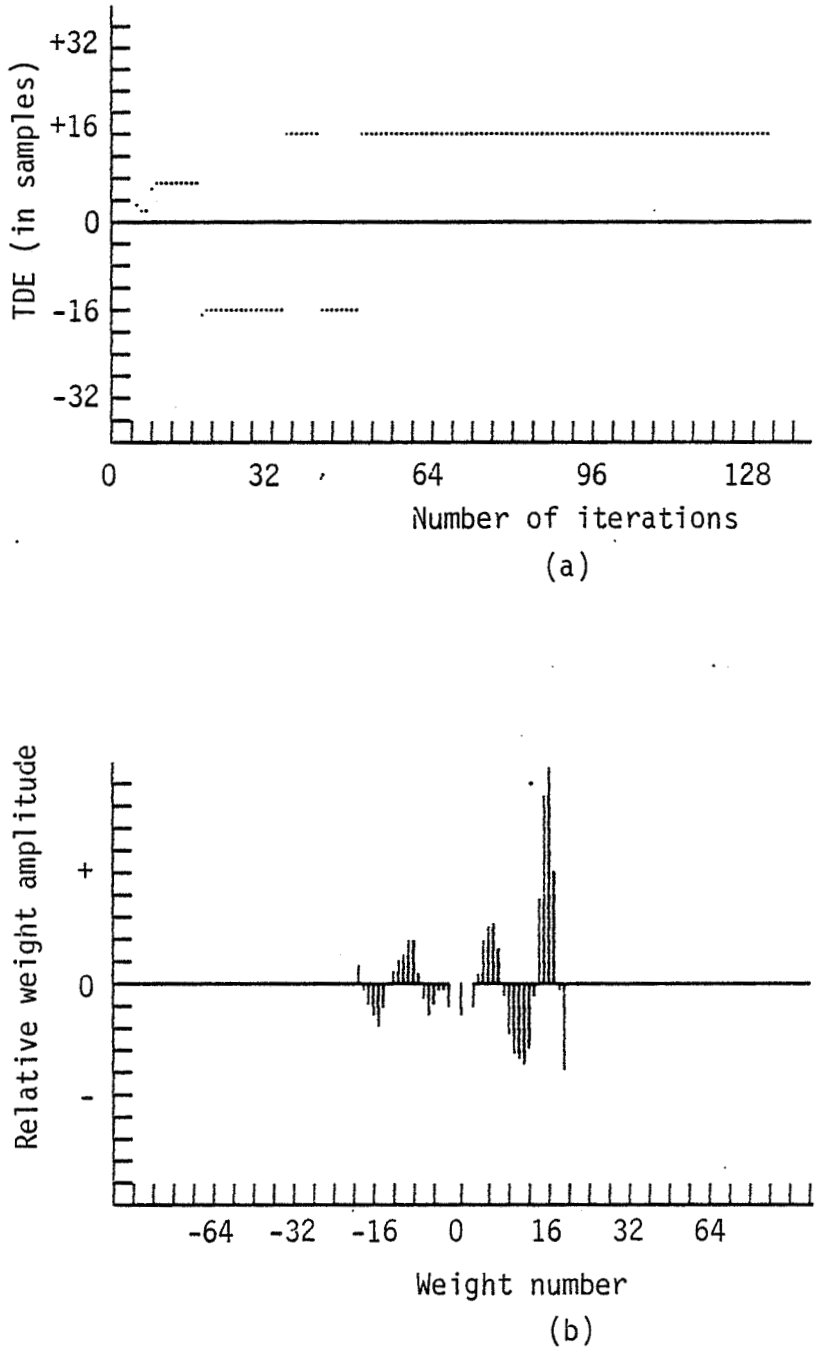


Figure 4.30. Filter response for jet data, C2 x C3, (a) TDE vs. time, (b) Final weight vector.

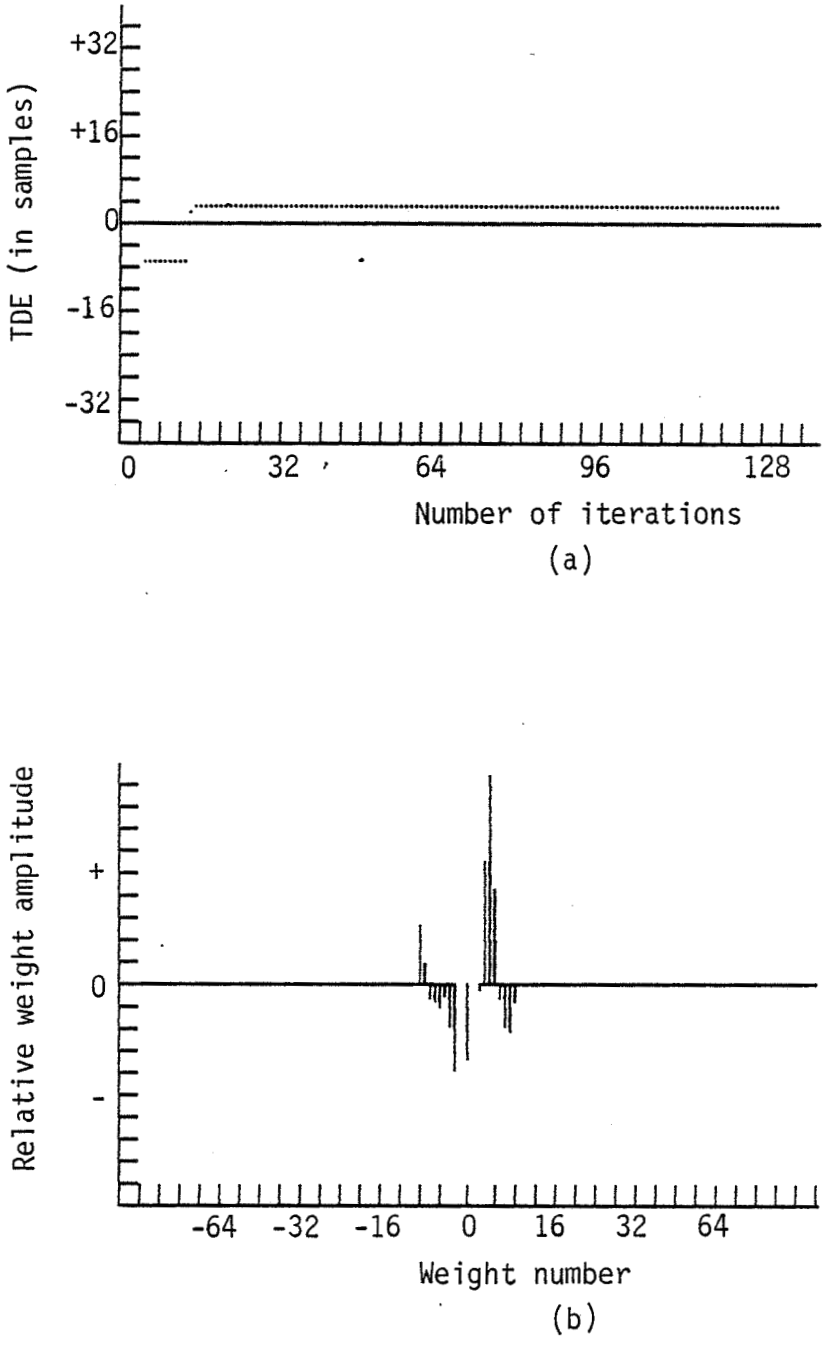


Figure 4.31. Filter response for jet data, C3 x C1, (a) TDE vs. time, (b) Final weight vector.

C2 leads C3 by 0.625 sec. (16 resolutions)

C3 leads C1 by 0.117 sec. (3 resolutions)

Based on these delays, a source direction of about 145° can be found. The estimate of the bearing places the source roughly on the middle of the runway as seen in Figure 4.25. This result among others obtained from similar files presents a reasonable assurance that the system is indeed capable of detecting and locating the sources of infrasounds.

4.5 Weather-Related Data

As was mentioned in Chapter one, the far-field meteorological events of interest in this research are clear-air turbulence and severe storms. There has not been clear evidence of clear-air turbulence detection on the system since it has been in operation. It is also believed that only a few storms were recorded. The uncertainty governing their detection is high in all cases except for one severe storm. This storm was the hurricane "DIANA." It was relatively close (200 miles) and stationary for at least one day. The data files stored by the system during that day, and their evaluation, present enougn necessary and reasonably sufficient conditions to identify the data with the hurricane. An overall evaluation of the system, including a discussion of the various factors that might have contributed to the lack of meteorological data, is discussed in the conclusion chapter. This section will concentrate on the hurricane and the issues surrounding its detection.

A tropical storm developed into a hurricane over the coastal water of North Carolina during the period of September 10 to September 12, 1984. It reached its full strength by September 12 with recorded wind velocity up to 110 mph. The location of its "eye" on that day was relatively stationary about 200 miles away from NASA/Langley Research Center at latitude 33.8 and longitude 77.3. During the afternoon and evening of the same day, the system recorded eleven files, four of which pointed toward the site of the hurricane with a calculated sound pressure level of 85 dB. The first three files, were stored within approximately a 40 minute time frame. This raises the possibility of having all three files associated with the same event. Examination of the spectral shape and contents of the signal on one channel through all four files supports this possibility. For reference, information in the signal sampled on one microphone is presented in Figure 4.32 as time history, in Figure 4.33 as auto-power spectrum, and in Figure 4.34 as log-amplitude vs. log-frequency auto-power plots.* Compared to other data files recorded, these data present three, somehow unique, features. First, the signal has a wide frequency band. This can not be seen on the linear scale of Figure 4.33, because of the difference in the dynamic range of the components on both ends of the spectrum. But it is clear on the log-scale of Figure 4.34 where the band plotted is 2-to-16 Hz. Second, the roll-off of the spectrum is consistent with the prediction of Meecham and Ford [21]. They predicted a roll-off proportional to $\omega^{-3.5}$,

*Software used to obtain these plots was developed at NASA/Langley Research Center, by Dr. Allan J. Zuckerwar and David Katzoff.

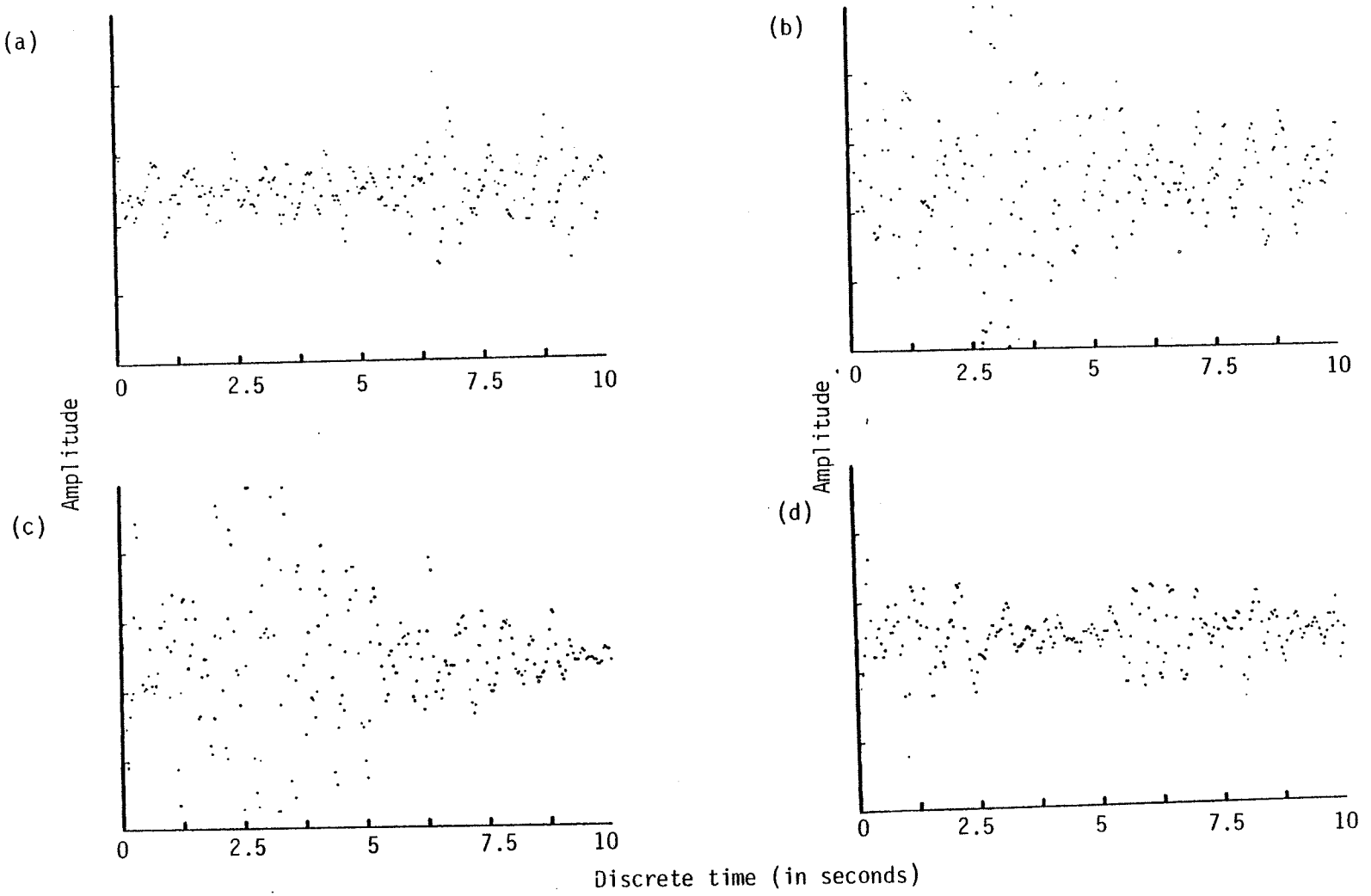


Figure 4.32. Time histories (in equivalent arbitrary units) of hurricane signal sampled on one channel at different times, (a) File one, (b) File two, (c) File three, (d) File four.

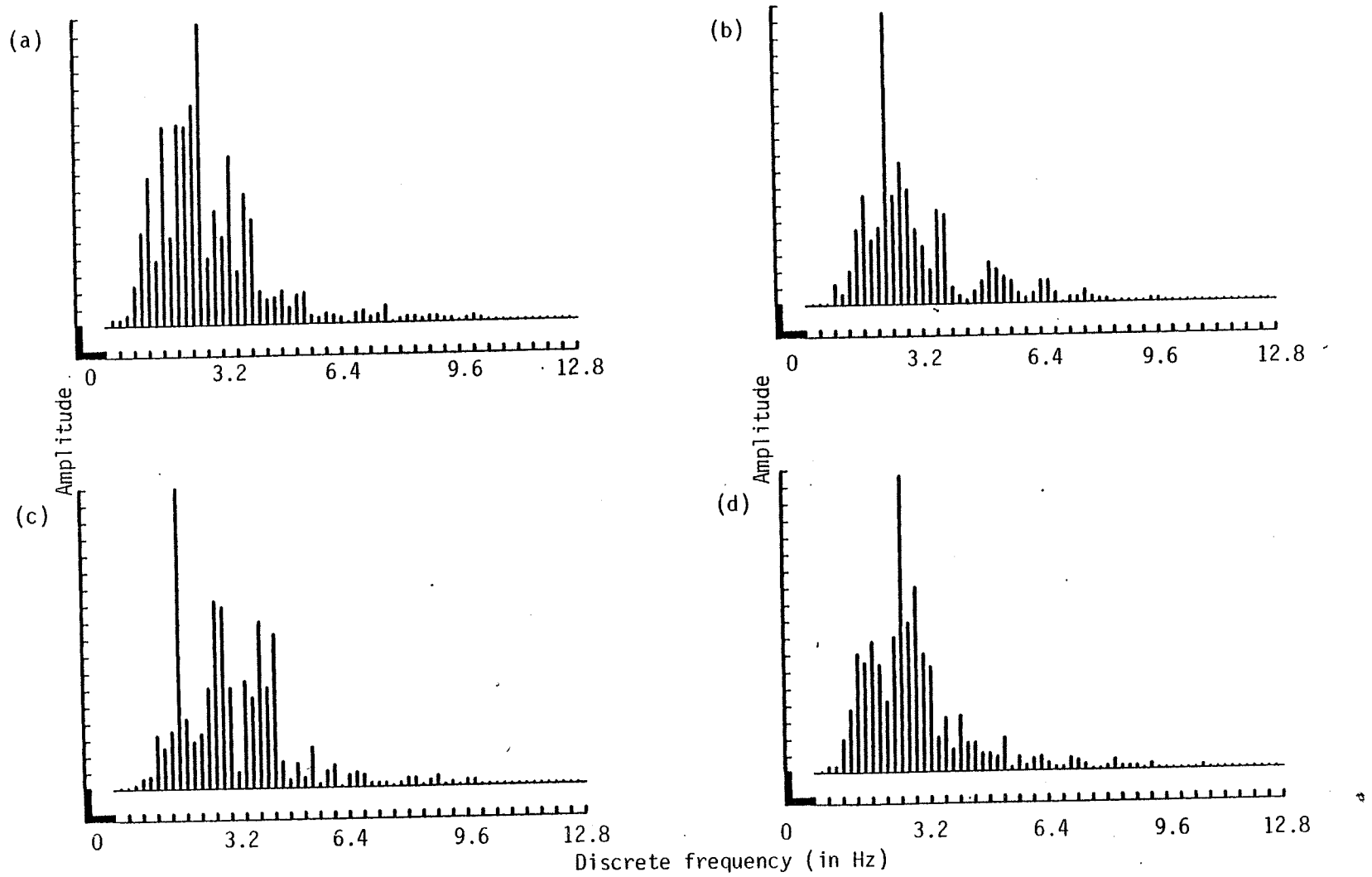


Figure 4.33. Auto-power spectrums (in arbitrary units) of hurricane signal sampled on one channel at different times, (a) File one, (b) File two, (c) File three, (d) File four.

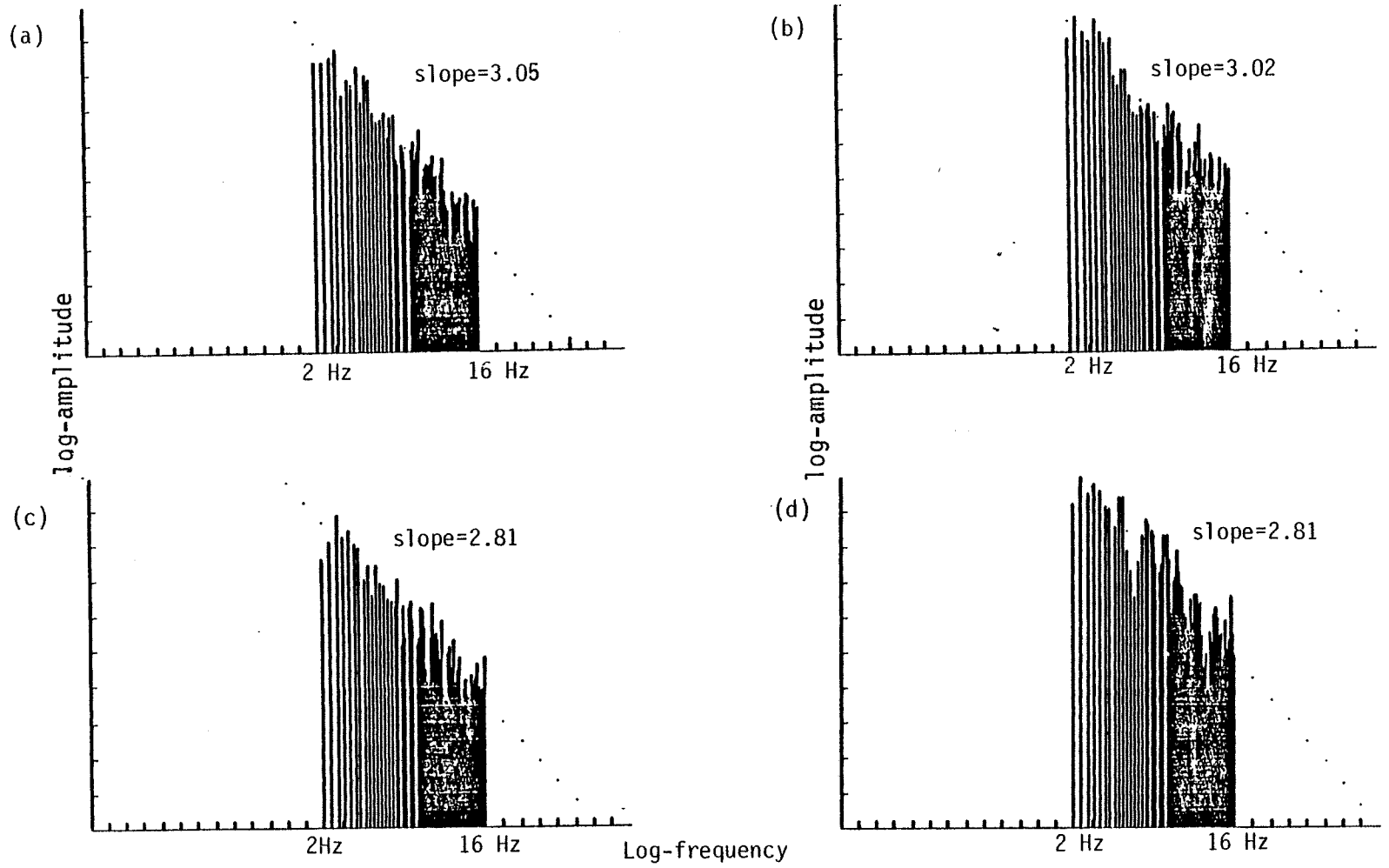


Figure 4.34. Auto-power spectrums on a log scale (log amplitude vs. log frequency, in equivalent arbitrary units) of hurricane signal sampled one channel at different times, (a) File one, (b) File two, (c) File three, (d) File four.

while the ones computed in Figure 4.34 range between $\omega^{-2.41}$ and $\omega^{-3.05}$. Third, the frequency band as well as the roll-off were consistent through all the files even though the fourth file was sampled hours after the first three.

To illustrate the consistency between the microphones outputs, the signal on all three microphones during the same time period was plotted in Figure 4.35. Its auto-power spectrum is plotted in Figure 4.36 on a linear scale and in Figure 4.37 on a log-scale.* The adaptive filter algorithm was reapplied to the data in Figure 4.35 to demonstrate the type of convergence obtained for this signal. The response is shown in Figure 4.38, Figure 4.39, and Figure 4.40 for C1 x C2, C2 x C3, and C3 x C1, respectively. As can be seen in the figures, the TDE vs. time plots reflect the noisy (actually windy) environment in which the signal traveled and was then sampled. The directions evaluated from the different four data files were, with respect to the bearing resolution, within an acceptable tolerance from each other. A mean direction of 187°, from north, was obtained. This bearing is to be compared with the actual value of 193° as illustrated in Figure 4.41.

The foregoing analysis and results clearly constitute the necessary conditions for true detection of the hurricane. The sufficiency, however, is lacking a higher detection rate, on behalf of the system, over a longer period of time. This issue, among others, will be discussed in Chapter 5.

---

*Software used to obtain these plots was developed at NASA/Langley Research Center, by Dr. Allan J. Zuckerwar and David Katzoff.

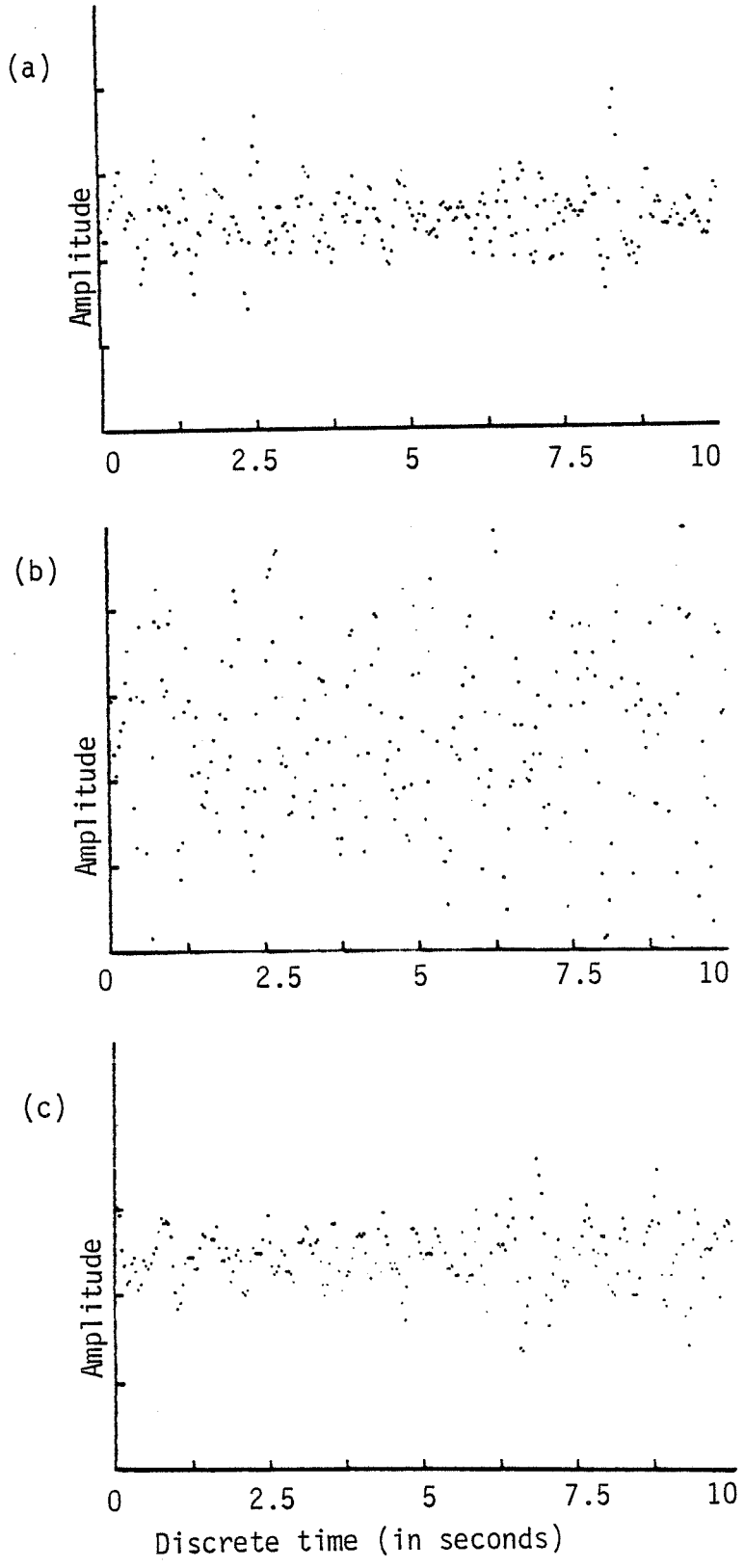


Figure 4.35. Time histories (in equivalent arbitrary units) of hurricane signal simultaneously sampled at (a) channel one, (b) channel two, and (c) channel three.

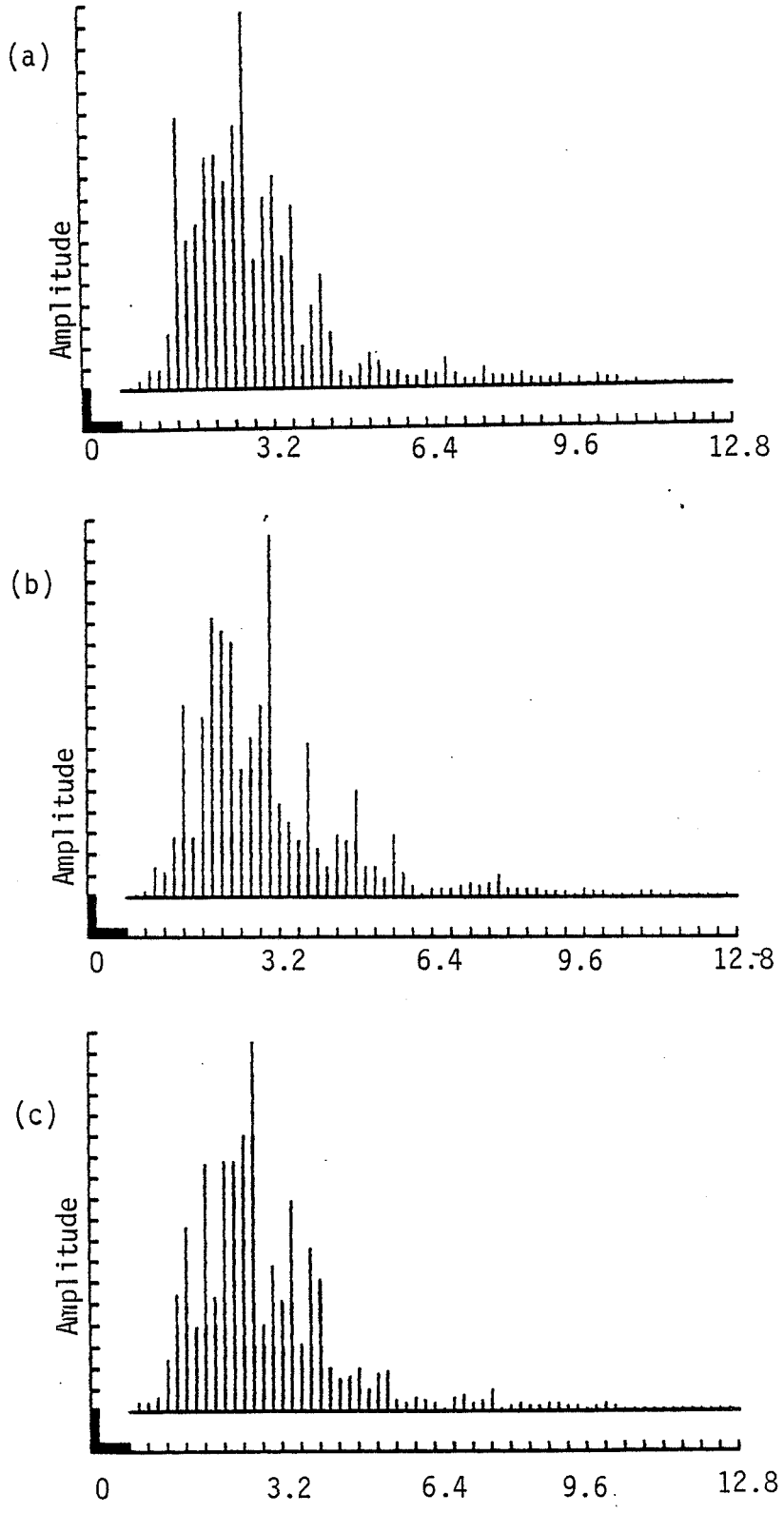


Figure 4.36. Auto-power spectrums (in arbitrary units) of hurricane signal simultaneously sampled at (a) channel one, (b) channel two, and (c) channel three.

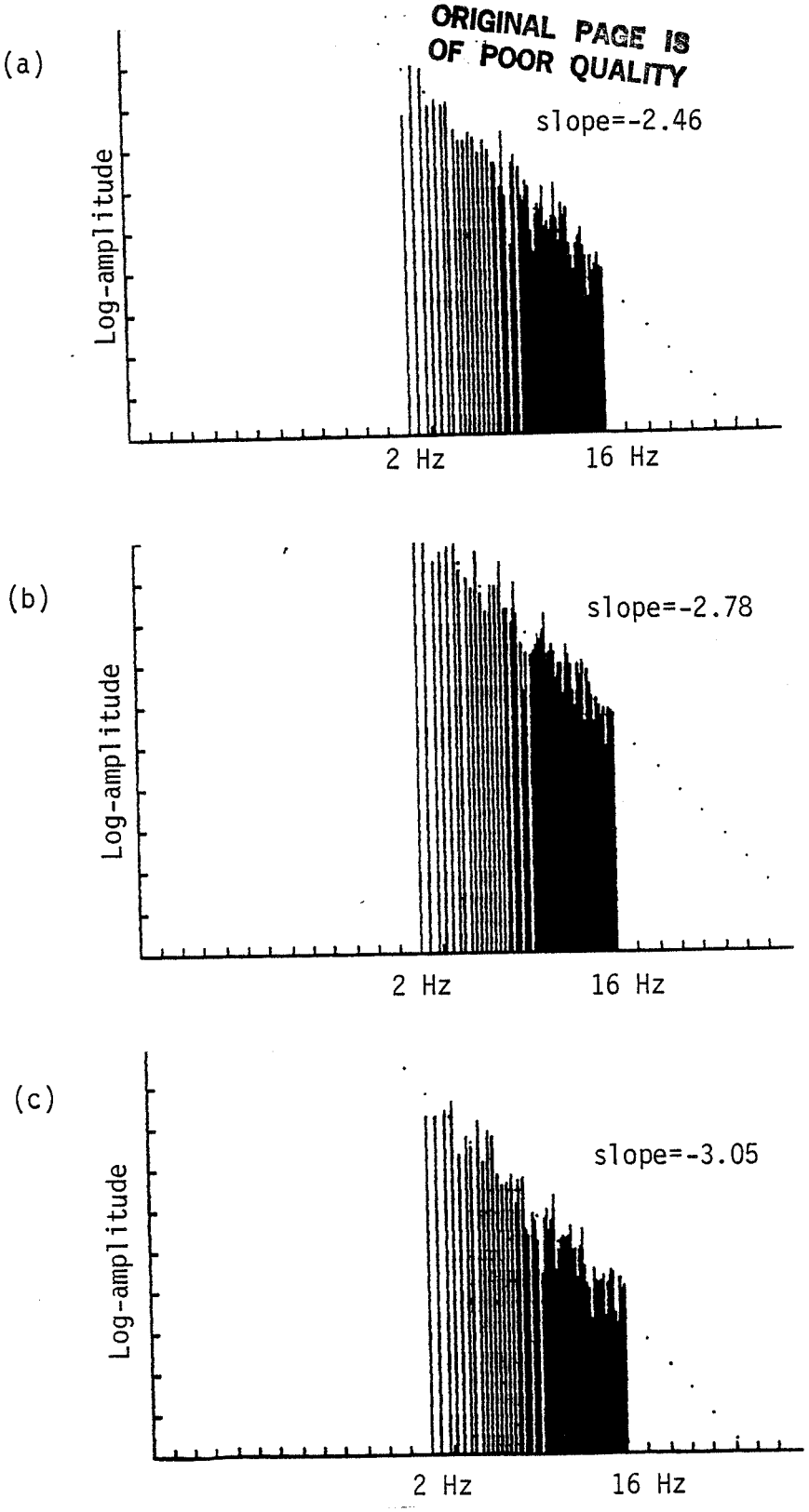


Figure 4.37. Auto-power spectrums on a log scale (log-amplitude vs. log-frequency, in equivalent arbitrary units) of hurricane signal simultaneously sampled at (a) channel one, (b) channel two, and (c) channel three.

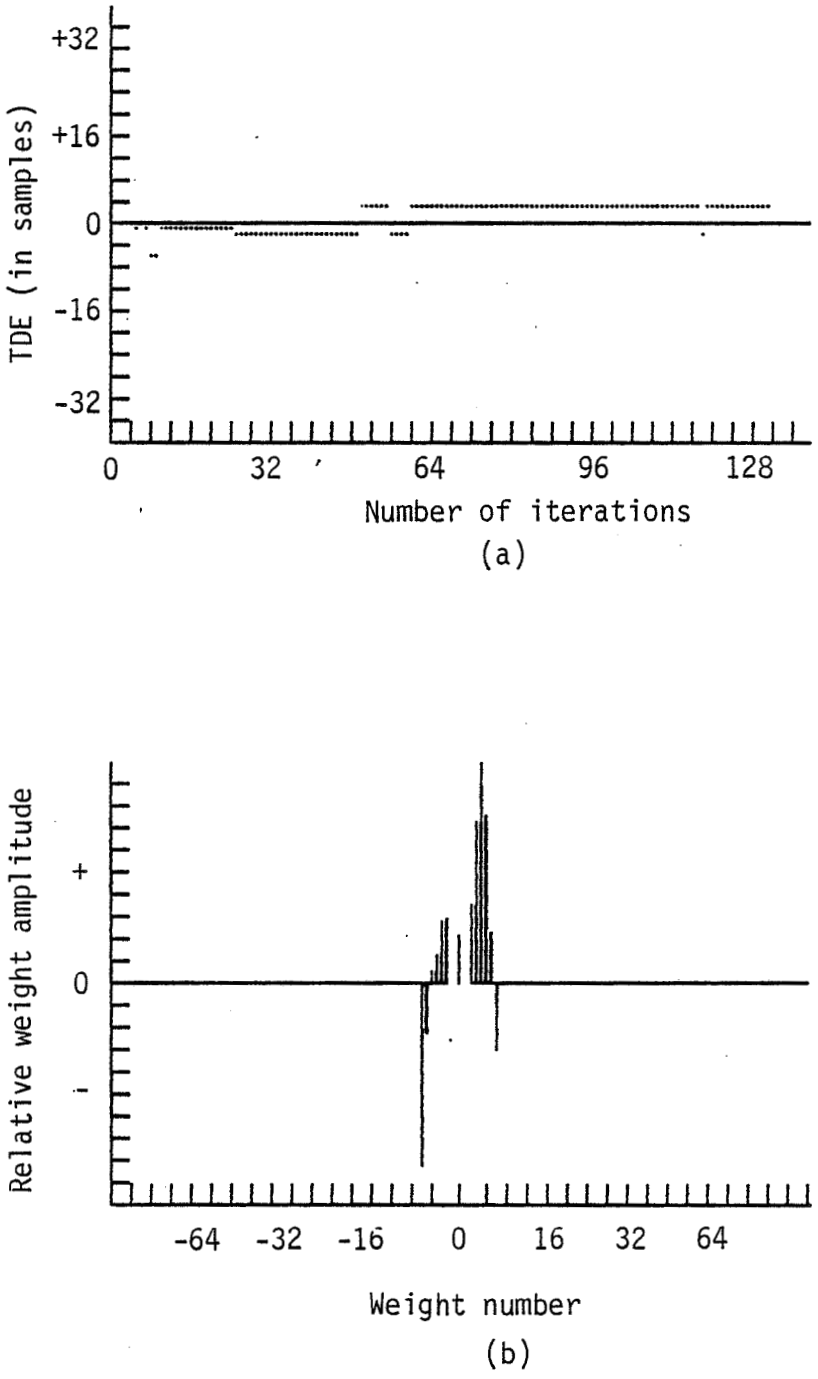
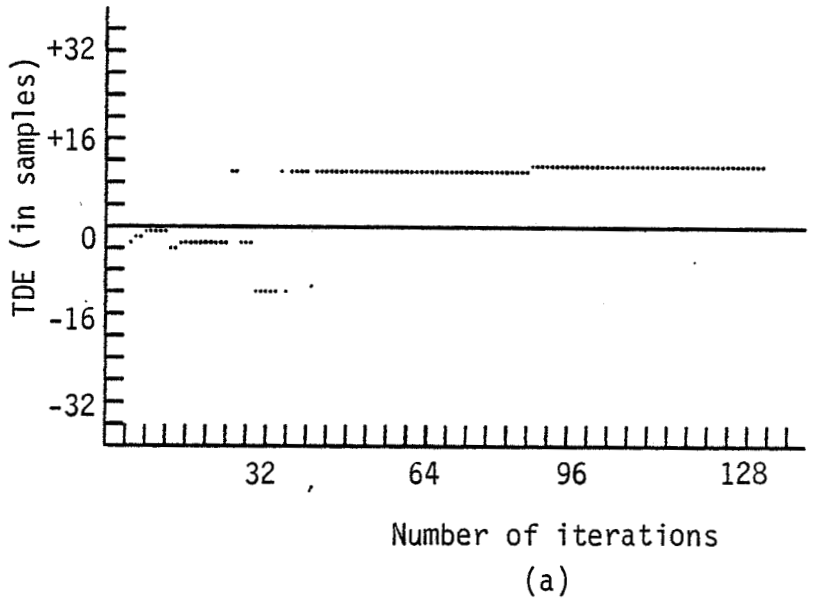


Figure 4.38. Filter response for hurricane data, C1 x C2, (a) TDE vs. time, (b) Final weight vector.



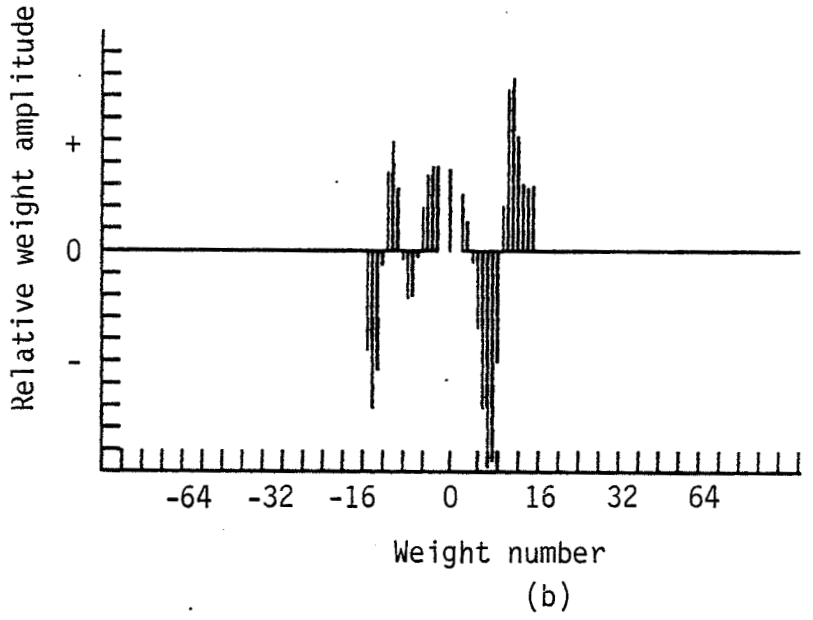
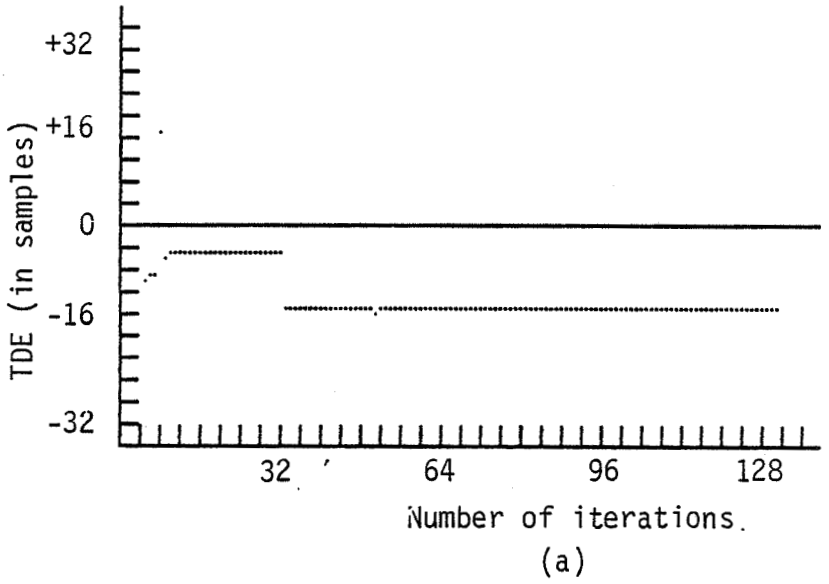


Figure 4.39. Filter response for hurricane data, C2 x C3, (a) TDE vs. time, (b) Final weight vector.



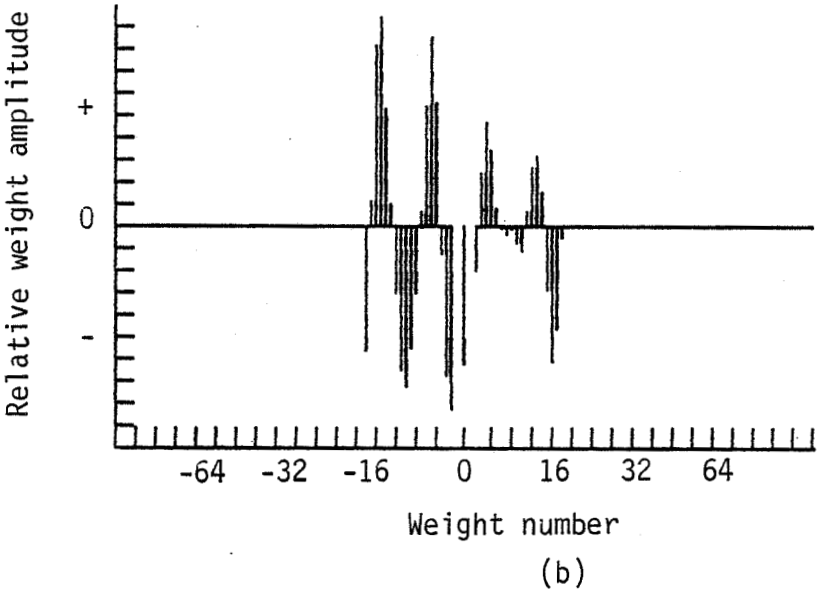


Figure 4.40. Filter response for hurricane data, C3 x C1, (a) TDE vs. time, (b) Final weight vector.

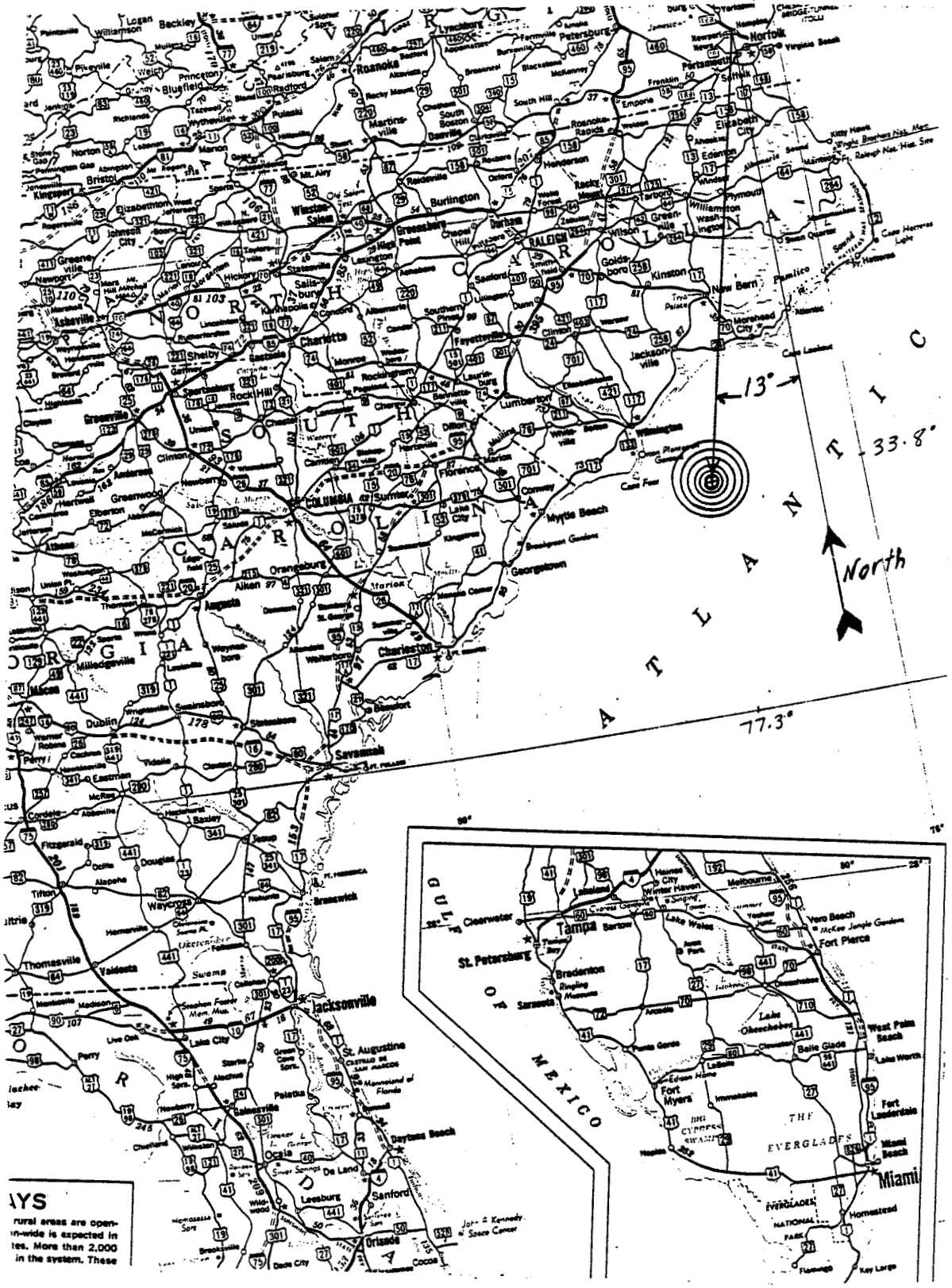


Figure 4.41. Hurricane site relative to the microphone array in NASA Langley Research Center.

4.6 Summary

The response of the basic LMSTDE algorithm was evaluated for deterministic data sequences. The simulation justified the specific choice of the filter parameters. The effect of varying the signal characteristics on the response was demonstrated. A direct proportionality between the SNR and the rate of convergence of the weight vector was observed. The dependency of the main lobe width of the weight vector on the frequency band of the signal, as predicted by Ahmed and Carter [17], was verified. The ability of the algorithm to track time-varying delays was also illustrated in the simulation. A steady state time-lag between the actual varying delay and the estimate of about 64 samples was observed for SNR=2. The algorithm was then tested using Space Shuttle VIII launch noise. The response was satisfactory in the sense that the launch site was pin-pointed within the bearing resolution. The specific software used in the field on a daily basis was described. It consisted of an automatic detection and bearing evaluation routine. The operation of the system over the period of June to September 1984 was evaluated with respect to the rate and type of signals detected. Data evaluation proved that man-made signals dominated the data files stored by the system. A sample of detected signals from jet-powered aircraft was presented. This sample verified the capability of the system to detect infrasounds and locate their sources. The system detection of weather-related infrasounds was not conclusive. Only one severe storm was detected with reasonable assurance. The assurance of the detection was based on the consistency of the signal characteristics and bearings through several files sampled on the day of the storm.

CHAPTER 5

CONCLUSION

5.1 Remarks

The goal of this research was the design of a system for detection and bearing evaluation of infrasounds. Based on the results presented in Chapter 4, it is believed that this goal has been achieved. The adaptive LMSTDE algorithm employed in the design was evaluated in section 4.1. Simulation results were in total agreement with theoretical expectations. The response of the algorithm to the shuttle noise clearly demonstrated its ability to estimate delays in cases of long-range infrasonic propogation. The automated-system performance, in the actual field, proved effective through detection and bearing evaluation of man-made signals. This class of signals consisted of several types of infrasounds including that of jet powered aircraft, whose signature is similar to what is anticipated in meteorological events. Acceptable performance of the system, as an infrasonic detector and bearing evaluator, was also achieved for large-scale weather events such as the hurricane "DIANA."

This work, however, is faced with the difficulty of evaluating the system as a turbulence detector for two reasons, both of which are beyond the designer's control. First, the data obtained in this case is poorly defined. That is, it is not repeatable, nor is it unambiguously verifiable. The second reason is the lack of a large number of weather-

related data files needed for the evaluation. As pointed out earlier, few meteorological events were detected on the system. Only the hurricane was identified with reasonable assurance. Some of the factors that might have contributed to the low detection rate of events, or of files within the same event, are listed below.

1. Low sound pressure level of weather-related infrasounds. A minimum level of 54 dB, derived from the system calibration, is required at the site of the array for the signal to be detected. The fact that the hurricane, despite being a powerful event, produced only 85 dB, raises the question of how big and how close the event should be in order to be detected. This question is left open for acoustical investigators.
2. Noisy environment. The presence of strong man-made signals at the array location makes it difficult for the system to identify small-scale weather events.
3. Wind effects. Small wind fluctuations at the array can be considered uncorrelated noise and usually do not affect the system response. However, this is not the case in windy conditions. Strong wind may correlate at two or more microphones, destroying the coherence property of the turbulence signal.
4. Multiple delay. In situations of scattered thunderstorms or large frontal systems, the signal sampled at the array is the superposition of infrasounds from more than one event. The multiple delay pattern embedded in this type of signal creates an ambiguity problem that can not be solved on the existing

system.

5. Short-term stationary signals. Although the lifetime of a meteorological event is relatively long (several hours in some cases), the period over which the turbulence, and hence the infrasonic emission, exists is virtually unknown. Therefore, the event may be easily missed in the case of turbulence that is stationary over a short time relative to the computational speed of the system; i.e. 20 seconds worth of data are sampled every 7 minutes (average computational time), so infrasonic emission for less than 7 minutes can be missed.

5.2 Future Work

More work is needed to improve the system performance for turbulence identification purposes. One work area is the investigation of signal processing techniques that are appropriate for elimination of non-weather data and removal of wind effects from the signal before applying the adaptive LMSTDE algorithm. Once enough knowledge is obtained about the infrasonic signature of turubulence, pattern recognition methods can be used to discriminate against non-weather data. Removal of wind effects may require the development of a wind-measurement system that is synchronized with the infrasonic detection scheme. The purpose of this system would be to provide the wind information needed for compensation in the system input.

Another future research consideration is the study and modification of the LMSTDE algorithm for supressing the side lobes in the weight vector and processing of multiple delay signals. Both issues are

important for avoiding any ambiguity in the weight vector peak that may be caused by the characteristics of the input signal.

Future work also needs to consider implementing the LMSTDE algorithm in hardware using dedicated signal processing chips. Hardware implementation would dramatically improve the computational speed, and hence, the detection rate, of the system in the case of short-term stationary signals.

REFERENCES

[1] "Low-Altitude Wind Shear and Its Hazard to Aviation," by National Research Council., National Academy Press, Washington, D.C., 1983.

[2] Airline Pilot (Technical Talk), Vol. 46, No. 4, April 1977.

[3] P. Kuhn, Clear Air Turbulence: Detection by Infrared Observations of Water Vapor Science, 196, pp. 1099-1100, 1977.

[4] Allan J. Zuckerwar and Harlan K. Holmes, "A Unified Acquisition System for Acoustic Data." NASA Technical Note TN D-8327, March 1977.

[5] C. H. Knapp and G. C. Carter, "The Generalized Correlation Method for Estimation of Time Delay," IEEE Trans. Ascoust., Speech, Signal Processing, Vol. ASSP-24, No. 4, August 1976.

[6] J. C. Hassab and R. E. Boucher, "Optimum Estimation of Time delay By A Generalized Correlator," IEE Trans. Acoust., Speech, Signal Processing Vol. ASSP-27, No. 4, August 1979.

[7] A. Papoulis, Probability, Random Variables and Stochastic Processes. New York: McGraw-Hill, 1965.

[8] G. M. Jenkins and D. G. Watts, "Spectral Analysis and Its Applications," San Francisco, CA, Holden-Day, 1968.

[9] P. R. Roth, "Effective Measurements Using Digital Signal Analysis," IEEE Spectrum, Vol. 8, April 1971.

[10] G. C. Carter, A. H. Nuttall, and P. G. Cable, "The Smoothed Coherence Transform (SCOT)," Naval Underwater Systems Center, New London Lab, New London, CT, Tech. Memo TC-159-72, August 8, 1972.

[11] B. Widrow, "Adaptive Filter, in Aspects of Network and System Theory," R. Kalman and N. DeClaris, Eds., New York: Holt, Rinehart and Winston, 1971, pp. 563-587.

[12] Steven A. Tretter, Introduction to Discrete-Time Signal Processing," New York: John Wiley and Sons, 1976.

[13] L. J. Griffiths, "Rapid Measurement of a Digital Instantaneous Frequency," IEEE Trans. Ascoust. Speech, Signal Processing, Vol. ASSP-23, April 1976.

[14] B. Widrow, et al., "Adaptive Noise Cancelling Principles and Applications," Proc. IEEE Vol. 63, December 1975.

[15] B. Widrow, P. Montey, L. Griffiths, and B. Goode, "Adaptive Antenna Systems," Proc. IEEE, Vol. 55

[16] F. A. Reed, P. L. Feintuch, and N. J. Bershad, "Time Delay Estimation Using the LMS Adaptive Filter Static Behavior," IEEE trans. Acoust. Speech, Signal Processing, Vol. ASSP-29, June 1981.

[17] D. H. Youn, N. Ahmed, and G. C. Carter, "An Adaptive Approach for Time Delay Estimation of Band Limited Signals," IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP 31.

[18] P. L. Feintuch, N. J. Bershad, and F. A. Reed, "Time Delay Estimation Using the LMS Adaptive Filter-Dynamic Behavior," IEEE Trans. Acoust. Speech Signal Processing, Vol. ASSP-29, June 1981.

[19] John A. McDonald, "Naturally Occurring Atmospheric Acoustical Signals," J. Acoust. Soc., Vol. 56, No. 2, August 1974.

[20] Eric S. Posmetier, "1-to-16-Hz Infrasound Associated with Clear Air Turbulence Predictors," Journal of Geophysical Research, Vol. 79, No. 12, April 20, 1974.

[21] W. C. Meecham and G. W. Ford, "Acoustic Radiation from Isotropic Turbulence," The Journal of the Acoustical Society of America, Vol. 30, No. 4, April 1958.

[22] M. Tahira and W. L. Donn, "Anomalous Infrasound from Space Shuttle and Skylab I," J. Acoust. Soc. Am., Vol. 73, No. 2, February 1983.

[23] C. S. Khalaf and J. W. Stoughton, "Software Development for Infrasound Measurement System," prepared for NASA Langley Research Center, under NAS1-17099-Task No. 10, August 1983.

APPENDIX A

This appendix represents the derivation of a relationship between the source bearing and source elevation, and the delays obtained for an arbitrary sound wave. A general relation is first derived for one pair of microphones. The specific relations for the array used in this work are then discussed.

Figure A.1 illustrates the elevation effects on the delay between two microphones, where

- θ is the source direction relative to the line joining the microphones
- ϕ is the source elevation angle
- D' is the delay due to the elevated-source signal
- D is the projection of D' on the horizontal plane (delay for $\phi=0$)
- V is the sound velocity
- ℓ is the distance between microphones.

Inspection of the figure yields the following two relations,

$$\cos \phi = \frac{D'xV}{DxV} = \frac{D'}{D} \qquad (A.1)$$

and

$$\cos \theta = \frac{DxV}{\ell} \qquad (A.2)$$

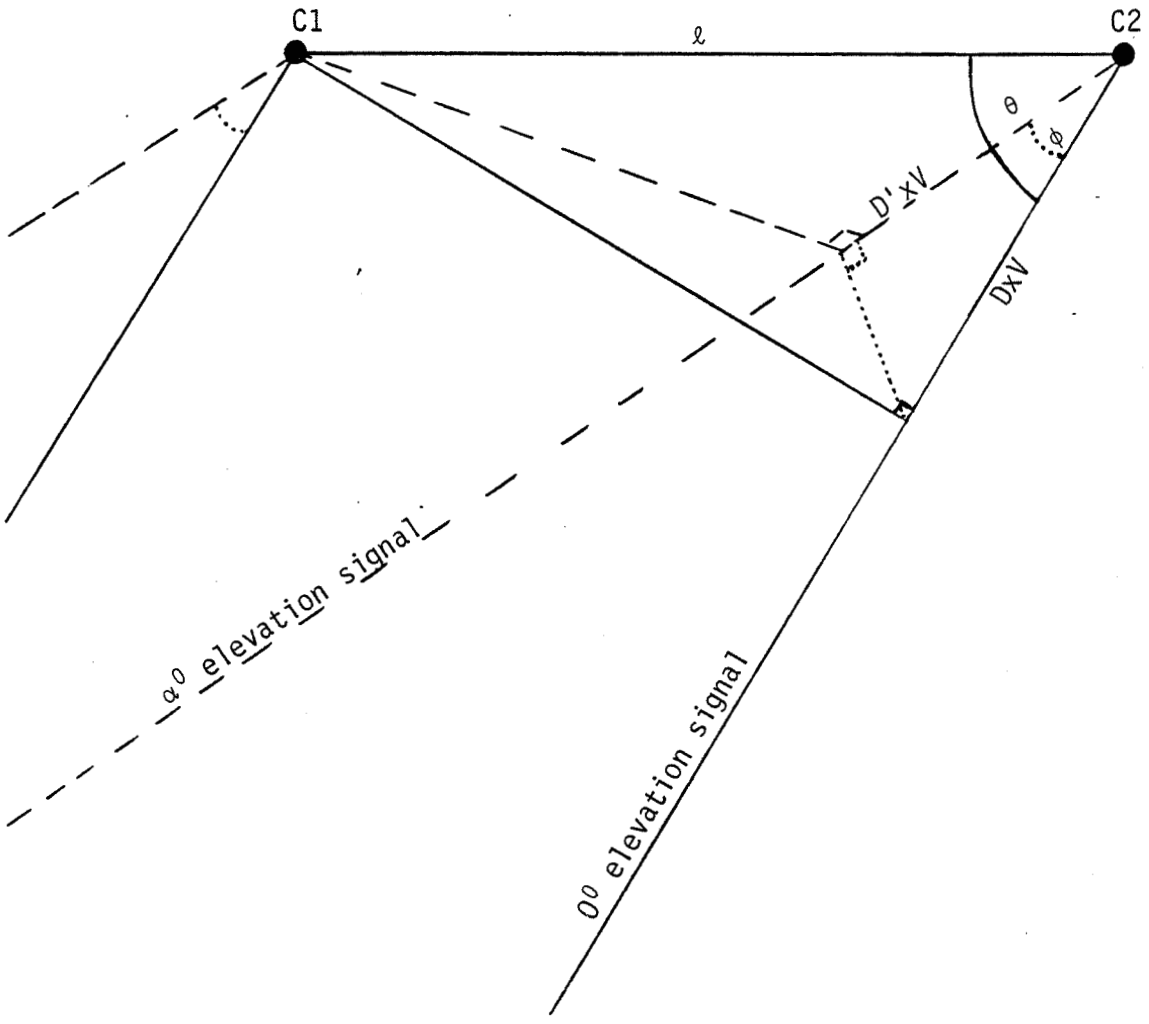


Figure A.1. Elevation effect.

substituting for D in (A.1) gives

$$\cos \phi = \frac{V}{\ell} \times \frac{D'}{\cos \theta} \qquad (A.3)$$

Equation (A.3) is a transcendental equation that relates the direction and elevation of the source to the delay between two microphones. Note that D' is given, as the output of the LMSTDE algorithm, while ϕ and θ are to be found.

The array configuration, and the values of θ $(\theta_1, \theta_2, \theta_3)$ for the three pair of microphones, are shown in Figure A.2 (horizontal plane). Let D_1', D_2', and D_3' be the delays obtained from microphones (3, 4), (3,2), and (3,1), respectively, due to a signal propogating with $\phi°$ elevation. Applying equation (A.3) to the different pairs yields

$$\cos \phi = \frac{V}{\ell} \frac{D_1'}{\cos \theta_1} \qquad (A.4a)$$

$$\cos \phi = \frac{V}{\ell} \frac{D_2'}{\cos \theta_2} \qquad (A.4b)$$

$$\cos \phi = \frac{V}{\ell} \frac{D_3'}{\cos \theta_3} \qquad (A.4c)$$

θ_1, θ_2, and θ_3 represent the source direction relative to the microphones axis as shown in the figure. If β represents the source direction relative to true North, then θ_1, θ_2, and θ_3 can be expressed

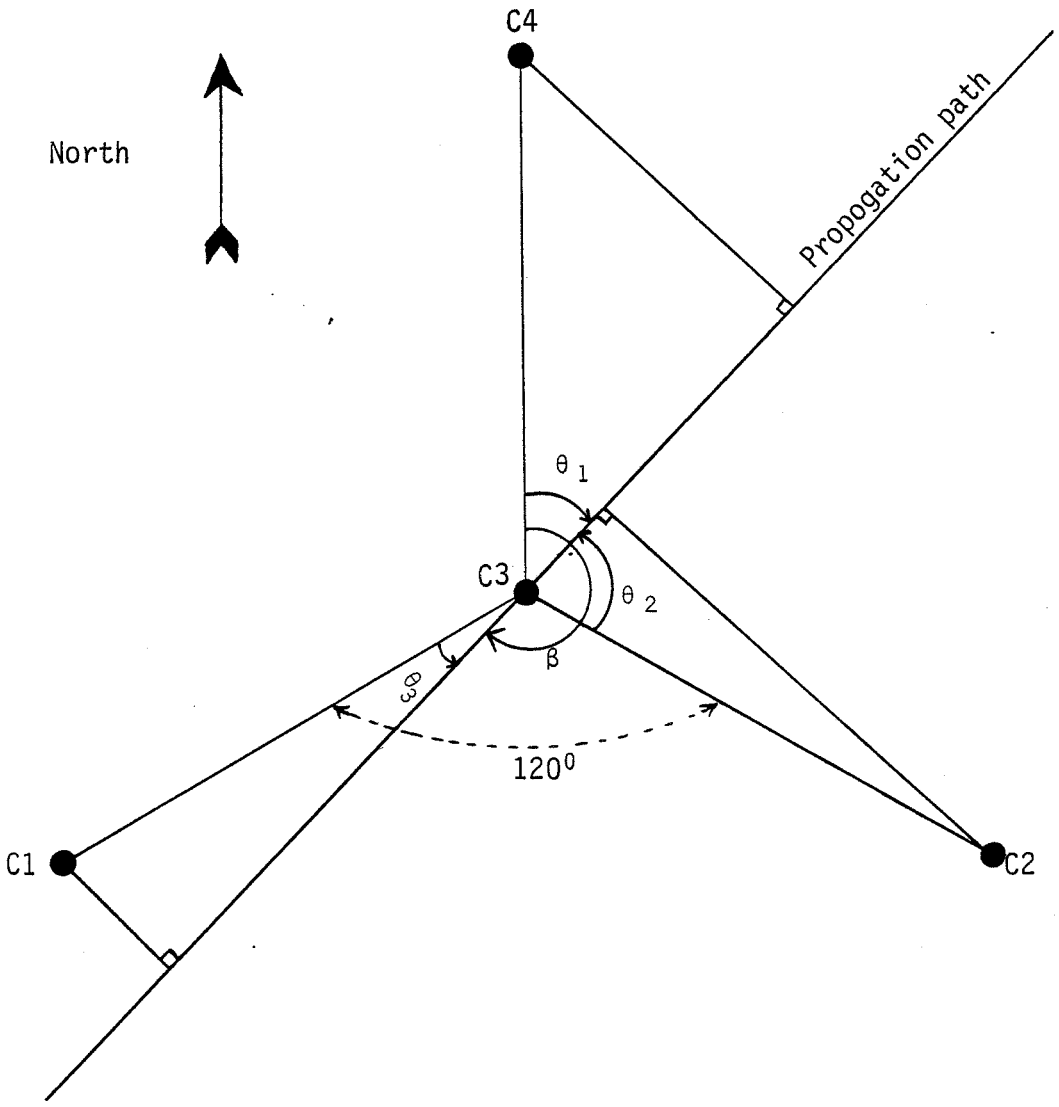


Figure A.2. Relative angles.

in terms of β by linear relationships, i.e., in figure A.2, the relations are

$$\begin{aligned} \theta_1 &= \beta - 180^\circ \\ \theta_2 &= 300^\circ - \beta \\ \theta_3 &= 240^\circ - \beta \end{aligned} \qquad \text{(A.5)}$$

Hence, having D_1', D_2', and D_3' estimated by the system, the problem of finding ϕ and β becomes that of solving two equations (out of the three in eq. (A.4)) with two unknowns.

Note that the relations of (A.5) hold for a certain range of β only. Thus, when solving for the elevation and the exact bearing of the source in a given situation, these relations have to be evaluated with the bearing values provided by the system.